1994年4月1日発行（毎月1回1日発行）第13巻4号通巻144号 昭和58年11月2日第三種郵便物認可

特集 SX-WINDOWの活用

Double Bookin'/EG Word/SX-BASIC/テキストマネージャ解析

決定！ 1993年度GAME OF THE YEAR

第6回アマチュアCGAコンテスト入選作品発表

4

1994

SOFTBANK
オー！エックス
定価600円

SHARP
目の付けところが、
シャープでしょ
夢の、頂きへ。
68ワールドの最高峰。
X68030
32bit PERSONAL WORKSTATION

32ビットパーソナルワークステーション

演算速度4.3倍(当社10MHz機比)/2.4倍(当社XVI比)※1、動画ウィンドウに見る新創造次元。選ばれた人だけが持つ感性によってX68030の扉はひらかれる。

X68000シリーズとして初の32ビットMPU MC68EC030を搭載して高速化を実現。

データキャッシュ、プログラムキャッシュをそれぞれ256バイト搭載したクロック周波数25MHzの高速32ビットMPUを搭載。演算速度は2倍以上上(当社従来比)※1の高速化を実現しました。また数値演算プロセッサMC68882※2(25MHz)もサポート。大量の実数演算を必要とするクリエイティブワークやGUI環境の操作性など、実行速度の飛躍的な向上が図られています。(当社従来比)

※1 Dhrystn(四則演算)比。25MHz・データキャッシュオン・プログラムキャッシュオンでMC68000/10MHz時の約4.3倍、16MHz時の約2.4倍。
※2 数値演算プロセッサCZ-5MP1標準価格54,800円(税別):本体内の専用ソケットに取りつけ可能。

65,536色表示、動画表示を実現。さらにパワーアップしたSX-WINDOWver.3.0。

X68000独自のウィンドウシステムとして定評の「SX-WINDOWver.2.0」をさらに強化した「SX-WINDOWver.3.0」を標準装備。新たに、65,536色の自然色グラフィック表示を可能とした『グラフィックウィンドウ』※を搭載。またアニメーション動画をウィンドウ上で表現でき、手軽にコンピュータアニメーションが楽しめる『CGAウィンドウ』、さらに従来のエディタのイメージを一新、高度な日本語文書作成をサポートするSX-WINDOW対応の高機能日本語マルチフォントエディタを標準装備。アウトラインフォントの展開もさらに高速化が図られています。

※SX-WINDOW上の512×512ドットのエリア内で表示可能。

GUIに対応する大容量メインメモリを搭載。

メインメモリは標準で4Mバイト、複数のアプリケーションをウィンドウ上で同時に使用するなど大量のデータ処理に対応。また本体内の増設で、I/Oスロットを使用せず最大12Mバイトまで拡張できます。拡張したメモリはすべて32ビットバスによる高速アクセスが可能、優れた拡張環境でシステムパワーアップをサポートします。

※メモリ増設には、4MB内部増設RAMボードCZ-5BE4標準価格54,800円(税別)、4MB増設RAMモジュールCZ-5ME4標準価格49,800円(税別)をご使用ください。なおCZ-5ME4はCZ-5BE4上に装着します。

X68000シリーズの高機能を継承した上で、さらに使いやすさの向上を図ったコンパチビリティ重視設計※1、すぐに使える高機能ソフトを標準装備。

●25MHzでは速すぎるアプリケーションも、従来のクロック周波数(10MHz/16MHz)で動作可能なソフトコンパチ重視設計●65,536色同時発色の自然色グラフィックス(最大表示エリア512×512ドット)、1024×1024ドットの実画面エリアを持つ高解像度表示能力(最大表示エリア768×512ドット)、疑似高解像度スーパーインポーズ(インターレース方式/512×480ドット・専用ディスプレイテレビ使用時)を装備した高精細度自然色グラフィックス機能。●外部MIDI音源もコントロール可能※2、ウィンドウ上で手軽にコンピュータミュージックが楽しめるMIDI音源対応デバイスドライバ搭載●ステレオ8オクターブ8重和音FM音源、ADPCM搭載●プリンタ、RS-232C、SCSI、オーディオ入出力、イメージ入力など多彩なインターフェイスを装備。●日本語変換効率や操作性を高めた日本語フロントプロセッサASK68Kver3.0搭載。●従来のエディタのイメージを一新したSX-WINDOW対応の高速多機能日本語マルチフォントエディタ標準装備●日本語マルチフォントエディタ中に貼り付ける絵やグラフなどが簡単に作成できるグラフィックパターンエディタ●MIDI対応のX-BASIC。

※1 アプリケーションソフトおよび周辺機器のうち、一部動作しないものがあります。詳しくはシャープお客様相談窓口にお問い合わせください。
※2 別売のMIDIインターフェイスが必要です。

130mmFD(5.25型) マンハッタンシェイプシリーズ

■X68000伝統のマンハッタンシェイプを継承 ■130㎜FDD(5.25型)2基搭載 ■80MBハードディスク内蔵(CZ-510C)※ ■マウス・トラックボール標準装備 ■ASCII準拠フルキーボード採用
※CZ-500Cには、80MB内蔵用ハードディスクドライブCZ-5H08/160MB内蔵用ハードディスクドライブCZ-5H16を用意しています。

本体+キーボード+マウス・トラックボール
130㎜(5.25型)FDタイプ
CZ-500C-B(チタンブラック)標準価格398,000円(税別)
HD内蔵 CZ-510C-B(チタンブラック)標準価格488,000円(税別)
14型カラーディスプレイ
CZ-608D-B(チタンブラック)標準価格94,800円(税別・チルトスタンド同梱)

90mmFD(3.5型) コンパクトシリーズ

■32ビットのハイパワーを凝縮したコンパクトフォルム ■2DD対応90㎜FDD(3.5型)2基搭載 ■80MBハードディスク内蔵(CZ-310C)※ ■マウス標準装備 ■コンパクトキーボード採用
※CZ-300Cには、80MB内蔵用ハードディスクドライブCZ-5H08/160MB内蔵用ハードディスクドライブCZ-5H16を用意しています。

Compact

本体+キーボード+マウス
90㎜(3.5型)FDタイプ
CZ-300C-B(チタンブラック)標準価格388,000円(税別)
HD内蔵 CZ-310C-B(チタンブラック)標準価格478,000円(税別)
14型カラーディスプレイ
CZ-608D-B(チタンブラック)標準価格94,800円(税別・チルトスタンド同梱)

for X68 Mac

ビデオグラフィックスの世界へ。

1,677万色対応※1の高速映像取り込み、動画・静止画の手軽なハンドリングが、新たなマルチメディアシーンを創造する。

NEW

テレビやビデオ、ビデオカメラ、ビデオディスクなどの映像をX68シリーズ、Macintoshシリーズ※2の画像データとして高速取り込みが可能なビデオ入力ユニットです。動画、静止画を簡単にハンドリングできるアプリケーションソフト「ライブスキャン」も標準装備。新たなマルチメディア展開でX68シリーズの可能性をさらに拡げます。●SCSIインターフェイス採用●MC68EC020(25MHz)搭載。

ビデオ入力ユニット
CZ-6VS1

※1 X68シリーズは、65,536色表示。Macシリーズは、機種により異なります。※2 対応機種は、MacintoshII以降の機種です。
●MacはMacintoshの略称です。
●Mac、Macintoshは米国アップルコンピュータ社の登録商標です。

CZ-6VS1の詳しい紹介や新作ソフト情報などX68シリーズの最新情報を満載。ちょっと役立つデータやプログラムなど、いろいろ楽しめるディスク情報誌をプレゼントします。

[EXEディスク]プレゼント

●EXE会員は、EXE会員番号と、90㎜(3.5型)/120㎜(5.25型)を明記の上、官製ハガキでお申し込みください。また、これからEXEクラブへ入会される方は、商品同梱のアンケートハガキに「EXEディスク希望」と明記の上、ご投函ください。(応募〆切 平成6年5月25日消印有効)
●EXEクラブに入っておられない方は、ソフトベンダー「TAKERU」での購入が可能です。(平成6年4月1日より2ヶ月間、予価200円)

●消費税及び配送・設置・付帯工事費、使用済み商品の引き取り費等は、標準価格には含まれておりません。

■お問い合わせは… シャープ株式会社 コンシューマーセンター 西日本相談室 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

資料請求券 X68030 Oh!X 4係

Oh!X

CONTENTS

特集　SX-WINDOWの活用

第6回アマチュアCGAコンテスト

決定！ 1993年度GAME OF THE YEAR

ジオグラフシール

ローテク工作実験室

ビデオ入力ユニットCZ-6VS1

●特集

●カラー紹介

●THE SOFTOUCH SPECIAL

●THE SOFTOUCH

〈スタッフ〉
●編集長／前田 徹 ●副編集長／植木章夫 ●編集／山田純二 豊浦史子 高橋恒行 ●協力／有田隆也 中森 章 林 一樹 吉田幸一 華門真人 吉田賢司 朝倉祐二 大和 哲 村田敏幸 丹 明彦 三沢和彦 長沢淳博 司馬 護 清瀬栄介 石上達也 柴田 淳 瀧 康史 横内威至 進藤慶到 ●カメラ／杉山和美 ●イラスト／山田晴久 江口響子 高橋哲史 川原由唯 ●アートディレクター／島村勝頼 ●レイアウト／元木昌子 ADGREEN ●校正／グループごじら

表紙絵：塚田　哲也

1994 APR. 4

E N T S

■広告目次

先が、面白くなる。

ウィンドウ環境のプラットホームを確立、SX-WINDOW ver.3.0

■実画面：1,024×1,024ドット、表示画面：768×512ドット

●この画面は広告用に作成した、機能を説明するためのイメージ画面です。また、各種アイコン等は、SX-WINDOW ver.3.0がもつ機能を使って作成したもので、標準装備のものとは異なるものもあります。
●本広告中のエディタで表示している文字のフォントはZeit社の、「書体倶楽部」のフォントを使用しています。

シャープ株式会社 ●お問い合わせは…コンシューマーセンター西日本相談室 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

資料請求券 ペリフェラル Oh!X 4係

に見たGUIの新展開。

❶マルチフォントエディタ編集例。文字ごとに文字種、文字の大きさの指定、修飾が可能で、イメージデータの貼り付けもOK。

❷CONFIG.SYSやAUTOEXEC.BATなどの編集に便利な「エディタ」モードの例。このように日本語マルチフォントエディタは、用途に合わせてカスタマイズできます。

❸❶の画面をプリンタで印字した例。対応プリンタも増えました。(カラー印刷は誤差分散により65,536色対応)

❹「パターンエディタ」で作成したデータを、背景に設定できます。

❺バージョンアップした日本語フロントプロセッサASK68K ver.3.0の辞書メンテナンスがウィンドウ上で可能。

❻アイコンデータや背景データを作成する「パターンエディタ」。文字の貼り付けなど、編集機能も一段とフレンドリーに。

❼オリジナルに作成したアイコンパターンの例。

❽512×512ドットの範囲内で65,536色の表示が可能。

❾さまざまなグラフィックフォーマットに対応しています。

❿任意のサイズに縮小・拡大表示可能。

⓫異なる画像フォーマットへのコンバートができます。

⓬「CGAウィンドウ」、65,536色(最大)のコンピュータアニメーション表示が可能です。

発展性のあるプラットホームとしてのウィンドウシステム、
SX-WINDOW ver.3.0が提供する新たなGUI環境が
さらなるウィンドウ時代を予見する――――。
国産オリジナルウィンドウとしての意味、未来への確かなビジョン、
ユーザーインターフェイスや高速化へのゆるぎない探求が
ここに凝縮されています。
65,536色表示はもちろん、さまざまな画像フォーマット対応、
イメージデータのコピー&ペースト、
動画、音楽/音声再生をサポートするマルチメディア環境。
そして、何よりもこれらが密接に連携して
統合的にハンドリングできるエキサイティングな環境を創造しています。
未来を照準に入れたウィンドウアーキテクチャ、
そのインテリジェンスがいよいよX68030/X68000シリーズで享受できます。

X68030

X68030 Compact

X68000 XVI

X68000 XVI Compact

●65,536色対応、動画ウィンドウ標準装備。

SX-WINDOW ver3.0 システムキット

CZ-294SS(130㎜FD)/CZ-294SSC(90㎜FD)各標準価格19,800円(税別)

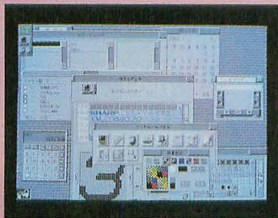
自然描画に迫る美しい表現が可能な65,536色表示のグラフィックウィンドウを装備。さらにグラフィックウィンドウ内でのアニメーション動画表示、各種グラフィックデータのコンバートも実現しました。またイメージデータの貼り付けなどをサポートした日本語マルチフォントエディタを始め、クリエイティブワークを支援する数々の便利機能を装備、Human68k ver.3.0システムディスクも付属しています。

※メインメモリ4MB以上必要です。

●「SX-WINDOW開発キット」のサポートツール。

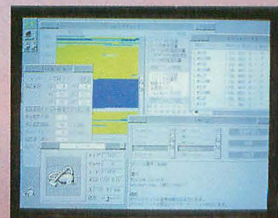
開発キット用ツール集

CZ-289TWD　標準価格12,800円(税別)

SX-WINDOW開発キットをさらに使いやすくするためのツールです。SXコールの簡易リファレンスを簡単に検索するインサイドSX、イベントの発生を常時監視確認するイベントハンドラ、リアルタイムにメモリブロックの利用状況を表示するヒープビューアなど11種のツールが用意されています。(2MB、ver.2.0)

●SX-WINDOW対応ドローイングツール。

Easydraw SX-68K

CZ-264GWD 標準価格19,800円(税別)

ホビーからビジネスまで幅広い分野で活用できる、待望のドローイングツールです。イラスト、フローチャート、地図、見取り図など各種グラフィックが製図感覚で作成できます。作成したデータは、他のSX-WINDOW対応アプリケーションでも利用でき、企画書やプレゼンテーション資料の作成をサポートします。またレーザープリンタドライバを付属、このドライバはSX-WINDOW対応の他のアプリケーションでも利用することができます。(4MB、ver.3.0)

●SX-WINDOWを楽しく使うためのアクセサリ集。

SX-WINDOW デスクアクセサリ集

CZ-290TWD　標準価格14,800円(税別)

SX-WINDOWをさらに便利に、楽しく使うためのデスクアクセサリ集です。スクリーンセーバ、アドレス帳、電子手帳通信ツール、パズルなど12種類の豊富なアクセサリが収められています。

1キーノート2スクリーンセーバ3スクラップブック4ミュージックボックス5ハイパーリンク(電子手帳通信ツール)6アドレス7スケジューラ8ウィンドウアイコニファイ9ソフトウェアキーボード10パズル11ファイルサーチ(ファイル検索ツール)12フォントリンカ。(2MB、ver.3.0)

●マルチタスク機能をはじめ、通信環境がさらに充実。

Communication SX-68K

CZ-272CWD　標準価格19,800円(税別)

通信環境をさらに高めたウィンドウ対応の通信ソフトです。マルチタスク機能により他のアプリケーションソフトを実行中でも簡単に通信が可能。また、ホスト局をクリックするだけの自動ログイン機能、初心者にも簡単なプログラム機能、最新モデム(20種類)もフルサポートしています。(2MB、ver.1.1)

●ウィンドウ対応グラフィックツール。

Easypaint SX-68K

CZ-263GWD　標準価格12,800円(税別)

マウスによる簡単操作、65,536色中16色の多彩な表現、クリエイティブマインドに応えるウィンドウ対応ペイントツールです。同時に複数のウィンドウを開いて編集でき、各ウィンドウ間でのデータ交換もできます。(2MB、ver.1.1)

●FM音源サウンドエディタ。

CZ-275MWD　標準価格15,800円(税別)

他のミュージックソフトで演奏中の音色を、簡単に作成、変更できるマルチタスク機能、またエディット、イメージ、ウェーブの3つの編集/確認モードを装備。作成中の音色も50曲の自動演奏でリアルタイムに確認、編集できます。まさにミキサー感覚で音創りが楽しめるツールです。(2MB、ver.1.1)

●SX-WINDOW対応になってさらにパワーアップ。

倉庫番リベンジ SX-68K ユーザー逆襲編

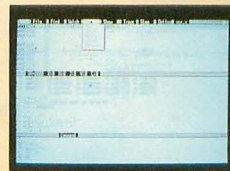
CZ-293AW(130㎜FD)/CZ-293AWC(90㎜FD)各標準価格6,800円(税別)

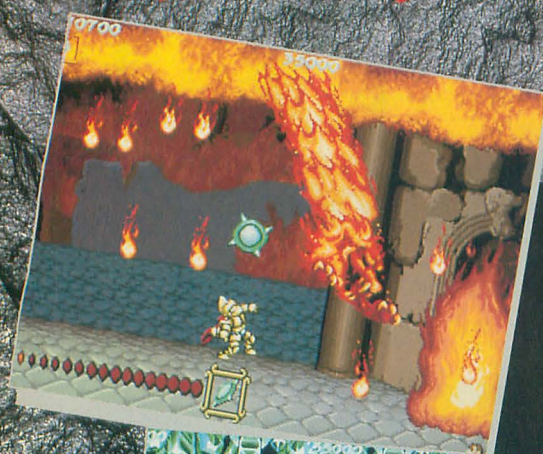
倉庫番10年にわたるユーザーの投稿など、新作306面が目白押し。まさに倉庫番の最強版がSX-WINDOW上で楽しめます。AI機能やエディット機能、キャラクタ変更機能も装備。半年で解けたらあなたは天才?です。(2MB、ver.1.1)

●X68030/X68000対応

CZ-295LSD　標準価格44,800円(税別)

※メインメモリ2MB以上が必要です。
※C compiler PRO-68K/ver.2.0/ver.2.1をお持ちの方には有償グレードアップサービスを行います。

C compiler PRO-68KのX68030/X68000対応版。MPU68030、MC68882の命令セットに対応したアセンブラ、デバッガ、ソースコードデバッガを付属。またHuman68k ver.3.0、ASK68K ver.3.0にも対応。新たにGPIBライブラリ、MC68882対応フロートライブラリを付属しています。

※(2MB,ver.1.1)の表示は、メインメモリ2MB以上、SX-WINDOW ver.1.1以上が必要であることを示します。

※EGWord、EGConvertは株式会社エルゴソフトの、一太郎は株式会社ジャストシステムの登録商標です。※発売予定のソフトの画面は実物とは異なる場合があります。

企業ユーザーのためのPC&WS情報誌

PCWEEK

PCWEEKは企業内PCキーマンから個人ユーザーまで広く読まれています。

PC&WSのニュースと特集記事を主体に、ワークグループ・コンピューティングに不可欠な総合的情報を提供する週刊誌です。
ハード、アプリケーションソフト、ネットワーク、基本ソフトなど全分野をどこよりも速く、詳しくお伝えしています。
パソコンの購入・導入をお考えの方、またビジネス戦略の武器として、ぜひPCWEEKのご購読をお薦めします。

一冊あたり200円でご購読できます

毎週金曜日発行（年間45回）
年間購読料金9,000円（税・送料共込）
タブロイド判／オールカラー／32〜64頁
米国Ziff Communications社提携
直接郵送制

書店ではお求めになれません

PCWEEKはすべて郵送制となっており、書店では購入できませんので、弊社に直接、年間定期購読でお申込ください。お申込方法は差し込みのハガキをご利用になって、今すぐお申込ください。
■お急ぎの方は電話／FAXでお申込みください。
TEL：0424-81-5510（平日9:00-16:00）
FAX：0424-81-5544

新規定期購読者好評募集中

どこよりも速い情報を毎週、直接机まで郵送します。

毎週読めば世の中の動きが良くわかる。
PCWEEKはPC&WS活用の新聞です。

最近のニュースから

■新98MULTiでMacやPS/Vを追撃■元アップルのジョン・スカリー氏21世紀の電話事情を講演する■PWIを推進するサン・マイクロ MSはX/Openの要請を拒否■Windows NTはNetWareを置き換えUNIXとの終末戦争にも勝てるか■NTの次に控えるCairo。核となるのは分散型オブジェクトファイルシステム■活用方法にユーザーの注目が移ってきたネットワークOS■インテルが隙間を埋める倍クロック486SXを計画中■速度重視とコストパフォーマンス重視でCPUアップグレードを比較してみる■3月に発売されるPowerPCマシンにグラフィック系のISVが虎視たんたん■急成長中の総合ソフト市場でMSを追撃するロータスのスイート戦略■ノベルがMacintoshクライアントに注力 ピアツーピアに至る4段階の拡張計画発表■テレビ会議製品のAPIが標準化へ■マルチベンダー環境下でトータルソリューション提供するウォロンゴング■PC OutlookConferenceでソフトベンダー4社首脳が語ったネットワーク■IBMがPowerOpen仕様の策定を目指してSMP対応AIXのベータテストを開始■オラクルがOracle7.1を発表 CDEやゲートウェイ製品も同時に■ノベルのUNIX事業の混乱はしばらく収拾の見込みなし

①期間中先着1,000名様にバックナンバー3号分プレゼント 1994.3/1〜1994.4/30まで

1年間の定期購読、通常45回が48回になるお得なチャンスです。ふるってご応募ください。
（新規購読の初回分郵送時に、その直前のバックナンバー3号分を同封してお送りする予定です。）

②毎月抽選で10名様にPCWEEKオリジナルバインダープレゼント

さらに毎月、抽選でPCWEEKの保管にたいへん便利なオリジナルバインダーがあたります。
※結果は商品の発送をもって代えさせていただきます。

ソフトバンク／出版事業部
〒103 東京都中央区日本橋浜町3-42-3
TEL 03-5642-8100

X68030 Inside/Out

[本書の概要]
- X68000とX68030の違い
- CRTコントローラ
- 数値演算プロセッサ
- エリアセット
- システムポート
- X68030の拡張スロット
- X68030の主要タイミングの実測結果
- 仮想記憶とMMU

桒野雅彦◉著

●B5変形判・X68030回路図付き●定価3,000円

X68000と比較しながら、
X68030のハードウェアの特徴を
本体内部と外部にわたって解析した本。
さらに、『Inside X68030』執筆後、
筆者が見つけた機能についても
補足的に説明してあります。
また、付録としてX68030にはついていなかった
MMU機能についての解説もしました。X68030の回路図付き。

X68k Programming Series #3

X680x0 TEX

続
刊

吉野智興/川本琢二/山崎岳志/実森仁志◉著

●B5変形判・2冊組・ビニール箱入り●5"FD7枚組●予価9,800円

待望久しいX680x0版TeXの解説書。
はじめてTeXに触れる人向けには簡単インストールプログラムを
付けてあるほか、すでにTeXを使っている人向けには
自分の環境にあわせたカスタマイズのしかたの説明もしています。
また、リファレンス編では
TeX、FONTMAN、PREVIEW.X、PRINT.X、MAKEFONT、
METAFONTの環境変数・オプションの説明などもあります。

●ディスク 1
install. x……………インストーラ本体
texmain. 1zh…………TeXを動作させるために必要なファイル群

●ディスク 2
texbin. 1zh…………¥TeX本体やフォントマネージャ、プレビューワ等の実行ファイル
mf. 1zh………………METAFONTを作成する場合に必要なファイルのすべてを収録

●ディスク 3
pk118. 1zh……………画面表示用の118dpi PKフォント
pk400. 1zh……………プリンタ用の400dpi PKフォント

●ディスク 4
pk180. 1zh……………プリンタ用の180dpi PKフォント
pk360. 1zh……………プリンタ用の360dpi PKフォント

●ディスク 5
pk240. 1zh……………プリンタ用の240dpi PKフォント
pk300. 1zh……………プリンタ用の300dpi PKフォント

●ディスク 6
freefont1. 1zh………フリーの日本語フォント
mincho46. bm1とmincho24.bm1

●ディスク 7
freefont2. 1zh………フリーの日本語フォント
gothic32. bm1とmincho32. bm1

＊なお、ディスクの内容は変更になる可能性があります。

SOFT BANK ソフトバンク株式会社出版事業部　〒103 東京都中央区日本橋浜町3-42-3 電話03(5642)8101

PⅣ スーパーリアル麻雀

原画&設定資料集

●(株)セタ/監修
●田中 良 /画

KAORI

A3描き下ろしポスターつき

YOU

ファン待望!! 初の公式ブック登場

アニメを見るかのような滑らかなグラフィックでそれまでの単なる麻雀ゲームの常識を打ち破った「スーパーリアル麻雀」シリーズ。PⅡではじめてかわいい美少女キャラが登場して以来、現時点での最新作PⅣまで、常にトップの人気を集めてきた。本書はそのPⅣの魅力を全部公開するはじめての公式設定資料集だ。アニメーターとして活躍中で、シリーズすべてのキャラクターデザインおよび原画を担当してきた田中良氏の血と汗と努力が結集した原画の数々と描き下ろしイラストをとくとご覧ください。

定価2,000円

3月下旬発売予定

MANA

祝！スーパーファミコン版3月25日発売決定！

■定価は税込 ■お求め・ご予約はお近くの書店で

ソフトバンク株式会社／出版事業部 販売局：TEL.03-5642-8101

GAME OF THE YEAR

恒例GAME OF THE YEAR。今年もたくさんのおはがきをありがとうございました。多くの大作が発表された1993年ですが，そのなかで最も素晴らしいゲームに贈られるOh!Xゲーム大賞をはじめとする各賞がとうとう決定いたしました。それでは，読者の投票で選ばれた作品とみなさんの声を紹介するとともに，1年間のゲームシーンを振り返ってみましょう。

[選択応募部門]

Oh!Xゲーム大賞

グラフィック賞
音楽賞
プログラミング技術賞
ゲームデザイン賞

[自由応募部門]

愛されたキャラクターたち
読者レポート　その他
勝手にGAME OF THE YEAR
ILLUSTRATION GALLERY
Oh!Xスタッフによる勝手にGAME OF THE YEAR

1993年度

GAME OF THE YEAR

受賞作品 悪魔城ドラキュラ

Oh!X ゲーム大賞

第1位 悪魔城ドラキュラ コナミ 295票

順位	タイトル	メーカー	票数
第2位	ストリートファイターⅡダッシュ	カプコン	206票
第3位	エトワールプリンセス	エグザクト	86票
第4位	餓狼伝説2	魔法株式会社	80票
第5位	コットン	EAビクター	62票
第6位	ロボットコンストラクションR.C.	エレクトリックシープ	42票
第7位	リブルラブル	電波新聞社	27票
第8位	ドラゴンバスター	電波新聞社	15票
第9位	ぶたさん	電波新聞社	14票
第10位	クレイジークライマー/クレイジークライマー2	電波新聞社	13票
	ネメシス'90改	SPS	13票

粒揃いのゲームのなかで大激戦が予想された1993年度Oh!Xゲーム大賞ですが，意外に大きく差がつきました。7月に発売されたコナミの会心作「悪魔城ドラキュラ」が，11月に登場したカプコンの大人気格闘ゲーム「ストリートファイターⅡダッシュ」の追い上げを振り切り，堂々の1位を獲得，1993年X68000ゲームの王座につきました。

思わず手が止まりそうになるほどの見事なグラフィック。いやがおうでも臨場感をかきたてる音楽に効果音。進むほどに新鮮な驚きの連続で，ますます引き込まれていく心憎いばかりの演出。これでもかと見せつけられる技術力。さらには機種の能力差やMIDI音源の有無など，ユーザー環境に応じた処理設定など，すみずみまで行き届いた配慮。すべての点において「完璧」を目指し，その結果，素晴らしいまでの完成度を実現したゲームといえるでしょう。

これまでにも数々のゲームで我々を楽しませてくれたコナミですが，1993年はこの

滴り落ちる血の涙は骸骨を生む。戦慄の美しさ

死神の大鎌が振り下ろされる。それは破滅の序章

傾く筏に容赦なく襲いかかる炎の揺らめき

この集計は，Oh!X1994年2月号のアンケートはがきより無作為抽出した1000件のデータによるものです。

ステンドグラスの輝きとは裏腹に，戦いは熾烈だ

まばゆい光のなか，邪悪な蝙蝠が舞い狂う

「悪魔城ドラキュラ」のみ。しかし，なかなか発表されない新作を期待と不安で待ち続けた多くのファンに対し，期待以上の大きな喜びをこの１本で与えてくれました。

さて，受賞は逃したものの第２位の「ストリートファイターⅡダッシュ」もX68000ゲーム史上に大きな足跡を残す作品です。カプコン参入時から熱狂的に待ち望まれ，経過すること１年半以上，それでも感動は色褪せずに大きな支持を受けました。

ほかにも上位には，バラエティに富んだ素晴らしい作品ばかり並んでいます。1993年のX68000ゲーム界は，寡作ながらも実り豊かな１年だったといえるでしょう。

読者のコメント

▶アクション性が高いし，演出も音楽もよい。１周目をクリアして再度プレイしたくなる。そして２周目はなつかしさでさらに感動。　奥村　真明（21）千葉県

▶グラフィック，サウンド，ゲーム性それになんといっても世界感。死神のステージが最高！　黒部　浩孝（21）三重県

▶楽々と感情移入できるグラフィックと音楽，楽しませてくれる数々のイベント。ムチの使い方が女王様なみになると敵の攻撃も心地よいものになるのだ。　安藤　広明（20）千葉県

▶死ぬと，X68000を蹴飛ばしたくなるほどくやしいぜ……ハァハァ。もう１回やろっと→また死ぬ→繰り返す。　高木　剛（21）福岡県

▶うちのマシンが030でなかったのがくやしくてしかたないくらいの，ハードを使い切った演出でした。ゲーム，グラフィック，サウンドのすべてがベストです。　小西　拓馬（18）滋賀県

▶壁が落ちるは，鏡に写るは，もーたいへん。　幸　俊威（19）大阪府

▶久しぶりにほかのマシンのユーザーにも「X68000はドラキュラができるぜ！」といばれるような作品だ。　佐藤　貴是（22）神奈川県

▶これはゲームではない。心地よい悪夢を見るための麻薬である。　中矢史朗（23）愛媛県

受賞のことば

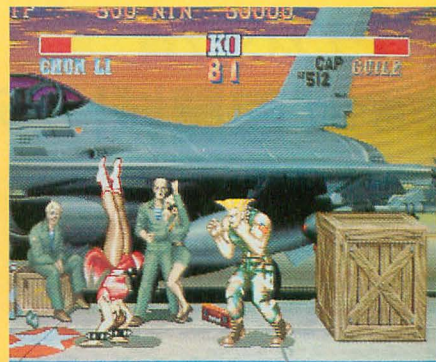
全国の投票していただいた皆様，ありがとうございました。思い起こせば，1991年に大賞をいただいた「パロディハウスだ！」で「来年１位の○ラ○○○○○」などと「ブラディウスⅡ」の受賞を予告しておいて，僅差とはいえ２位に甘んじてしまいました。で，「○ラ○○○○○」は「ドラキュラ悪魔」と読み替えてください（苦しい？）。

さらに今回は大賞に加え，部門賞もいくつもいただき，スタッフ一同，涙と鼻水まみれで喜んでおります。それでは受賞した各部門のスタッフからひと言。

・まずグラフィックから。ありがとうございまーす。デザイナー冥利につきます。やりたい放題の絵を描け，X68000は本当に素晴しい機種です。開発も楽しく，おまけにこんな賞をいただけるなんて，涙涙の私でございます。生きててよかった。ウーン腕がなるムチが鳴る。ヒュンヒュン。

・サウンドでんがな。X68000ユーザーの皆さん最近はどうでっか。あてらが３バージョンで苦しんだ甲斐がありまんがな。特にMT-32の評判がええ感じなので音色転送なんかで工夫したんが評価されたんでんなぁ。

・そしてプログラムから，ちょうどX68030の発売時期と重なり，当初の予定よりたくさんの特殊効果を入れることができました。ドラキュラ前や倒したあとのあれは10MHz機だけがターゲットならば没にしていたかもしれません。よかった，よかった。

あ，それからまだクリアしてない皆さん。ドラキュラ城内でもそろそろ冬物の絵から春物の絵へと衣替えをしている頃です。がんばってクリアしてくださいね。（コナミ）

第2位　ストリートファイターⅡダッシュ

日本中のX68000ユーザーの自宅をゲーセン化させた話題作。細かい部分まで完全移植でみんな大満足。変換アダプタで６ボタンプレイも可能に。

第3位　エトワールプリンセス

1993年前半の話題を独占したお姫さま，リルル。ぴょんぴょん跳んで単に可愛いだけじゃないのがエグザクトのセンスと技術力のすごいところ。

第4位　餓狼伝説2

ストⅡよりひと月遅れで発売という不利な状況で善戦。X68000オリジナルの四天王対戦も嬉しいところ。専用パッドのおまけも太っ腹でマル。

第5位　コットン

魔法使いコットンと妖精シルクのやりとりが楽しいシューティング。なんといっても「いっくぽーん」なのだ。誰でも楽しめる難易度設定も好評。

グラフィック賞

第1位 悪魔城ドラキュラ

コナミ／385票

順位	タイトル	メーカー	票数
第2位	ストリートファイターⅡダッシュ	カプコン	119票
第3位	エトワールプリンセス	エグザクト	64票
第4位	餓狼伝説2	魔法株式会社	63票
第5位	コットン	EAビクター	34票

音楽賞

第1位 悪魔城ドラキュラ

コナミ／377票

順位	タイトル	メーカー	票数
第2位	エトワールプリンセス	エグザクト	83票
第3位	コットン	EAビクター	72票
第4位	餓狼伝説2	魔法株式会社	34票
第5位	リブルラブル	電波新聞社	26票

プログラミング技術賞

第1位 悪魔城ドラキュラ

コナミ／215票

順位	タイトル	メーカー	票数
第2位	ストリートファイターⅡダッシュ	カプコン	162票
第3位	エトワールプリンセス	エグザクト	109票
第4位	餓狼伝説2	魔法株式会社	74票
第5位	コットン	EAビクター	34票

ゲームバランスのよさなど総合的な評価の高さでOh!Xゲーム大賞に輝いた「悪魔城ドラキュラ」ですが，これらの各賞でも圧倒的な強さをみせました。得票数もそれぞれで2位以下を大きく引き離す独走状態です。「Oh!Xゲーム大賞は好みの作品にし，グラフィックや音楽はドラキュラに」という人もいれば，逆に「大賞はドラキュラこそふさわしいが，ほかはそれぞれ好みの作品を選ぶ」という投票もあり，いずれにしても無視できない大きな存在だったようです。あらゆる人からまんべんなく支持を集めたのがこの素晴らしい結果となりました。

これらの賞は，いわばゲームのある一面に対してそれぞれ贈られることになるわけですが，それを独占してしまったこの「悪魔城ドラキュラ」。いかにそつなく，丁寧に作り込んであるかがうかがえます。

第5位までをみると，グラフィック賞，プログラミング技術賞はOh!X大賞とまったく同じ順位となりました。音楽賞のみは，他の賞で常にトップを追っていた「ストリートファイターⅡダッシュ」の評価が低く，第5位に食い込んだのは「リブルラブル」。なかでも「結婚行進曲」が評判です。

現在を刻む時計もやがて破壊のときを迎える

世界は現実と呼応し，循環する。季節は巡りくる

幻影はしょせんまぼろしにすぎない。しかし……

それにしても，このように各賞がほぼ同じ結果というのは，どの作品も総合的にバランスよく作られているということだと思われます。ますますハイレベルになっていくゲームの今後が楽しみですね。

読者のコメント

▶1993年の夏の夜を素晴らしいビジュアルと音楽そして演出で熱くしてくれた。
露崎　秀明（21）千葉県

グラフィック賞

▶手抜きなどという言葉の入る余地がないほどのグラフィック。恐れ入りました。視覚的効果が高い場面が多いのも気に入ってます。　北村　聡宰（21）富山県

▶やっぱり「うしみつどき」にやると緊張する。　山崎　康則（19）北海道

▶いつも思うのだが，コナミのグラフィックは「黒」の使い方が非常に巧みだ。
山崎　拓人（19）三重県

音楽賞

▶music modeにしてナイスな曲を聴いています。　森山　裕史（20）埼玉県

▶夏，東京に行ったとき，秋葉原で「ヴァンパイアキラー」を聴いて鳥肌がたった。
藤原　正浩（16）福岡県

▶ブロック7のあの雰囲気に「もうやめてくれ！」と叫ばずにはいられなくなる音楽。
間宮　義晴（19）山形県

プログラミング技術賞

▶噴水の水からとびだす水玉の動きなど，素晴らしい効果を作り出している。
中沢　一夫（29）群馬県

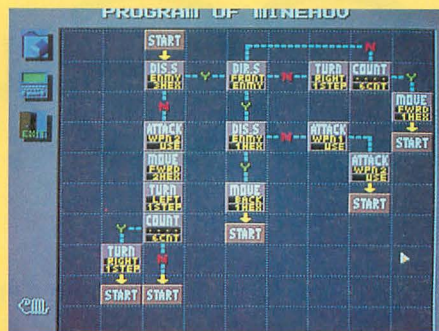
ゲームデザイン賞

第1位

ロボットコンストラクションR.C.

エレクトリックシープ／148票

第2位	悪魔城ドラキュラ	コナミ	86票
第3位	エトワールプリンセス	エグザクト	75票
第4位	リブルラブル	電波新聞社	70票
第5位	コットン	EAビクター	65票

今回の集計で最も票が割れたのが，このゲームデザイン賞でした。ここでも「悪魔城ドラキュラ」強し！　しかし，それを抑えてトップに立ったのは「ロボットコンストラクションR.C.」でした。エレクトリックシープ初の作品で堂々の受賞です。思い入れ度も高いようで，寄せられたメッセージも多く，「とにかく賞をあげたい」との声も。バトル大会も開催されるなど，開発者とユーザーの気持ちが一体となった，X68000らしい作品といえるでしょう。

フローチャートで動きを自在にプログラミング

対戦場所によっても勝ち負けは変わってくる

読者のコメント

▶まさにロボットRを設計Cする感覚を味わえた。お手軽なところも気に入った。
武田　和凡（20）京都府

▶最終的には人間VS人間のゲーム。人と人とのコミュニケーションが感じられます。
常木　修一（37）富山県

▶絶対に勝てるロボットが存在しない。バランスがいい。個人的にソフトハウスを支援したい。　小野　美樹夫（22）東京都

▶X68000をもっていない人までもがこれをもっている友人宅へ通い続けるほど新鮮なゲーム。　上原　誠（20）東京都

▶ロボットバトルものでいままでとっつきにくかった部分を見事にクリアしたゲームデザインには感嘆しました。しかも，奥が深い！　後迫　浩一（33）神奈川県

▶ある意味でパソコンゲームの究極の形であるといっても過言ではないと思う。
八木　真人（20）京都府

▶NIFTY-Serveでハマりました。既存のゲームデザインにとらわれないセンスに脱帽！このゲームが遊べるX68000ユーザーは幸せだと思います。山田　俊英（26）東京都

▶X68000オリジナルだということを高く評価したい。部品によるフローチャートというアイデアは見事。クリエイターのためのゲームだと思う。
小川　伸輔（19）宮城県

受賞のことば

まさか賞がもらえるとは思ってもいませんでした。投票してくださった方，どうもありがとうございました。私たちがゲームにおいて最も本質的だと思っている部門での受賞というのがなにより嬉しいです。当初，本数が伸びずに苦しみましたが「このゲームのためにX68000を買いました」とか「いままで遊んだゲームのなかでいちばん面白かった」などのはがきをいただき，ずいぶんと救われました。大会も郵送や通信などで何度か行うことができました。これからも自分たちの本当に創りたいゲームを創り続けていきたいと思います。
（エレクトリックシープ）

読者のコメント　1等賞はとれなかったけれど……

Oh!X大賞第6位　ロボットコンストラクションR.C.
▶徹夜した時間がいちばん長いソフト。買って半年たつけどいまだに徹夜している。
三好　俊夫（29）佐賀県

グラフィック賞第2位　ストリートファイターⅡダッシュ
▶リュウとケンの垂直中or小キックのときのズボンの穴まで再現している。
三浦　浩一（19）新潟県

音楽賞第2位　エトワールプリンセス
▶森で鳥が鳴いたりと，ただ音楽するのではない音というのも重要だと思った。
三谷　夏樹（24）千葉県

プログラミング技術賞第3位　エトワールプリンセス
▶高さの概念がとてもよくできていたと思います。ゲーム自体はちょっと短かったけど，面白かったです。　武藤　一文（21）埼玉県
▶ドラキュラもすごかったけど，やはり電源オンですぐ起動のエトプリに軍配が上がりました。技術の見せ方もいやみでなく，ゲームをしていて違和感がありませんでした。
加藤　和人（18）愛媛県

プログラミング技術賞第4位　餓狼伝説2
▶四天王が使えるし，移植も早かった。大きいキャラがちゃんと動く。荻原　亮（18）埼玉県
▶2ライン，避け攻撃など，ストⅡよりも大変な処理（と思う）をしていても，結構動く。
平山　悟（20）福岡県

プログラミング技術賞第5位　コットン
▶起動したらすぐにコミカルな音楽と，それに合わせてキーが光る。こんな凝った演出も珍しい。プレイする前から楽しい気持ちにしてくれた。　北川　修（20）奈良県

プログラミング技術賞選外　極
▶とにかく強い。冷や汗もんだ。
大塚　雅嗣（21）宮城県

ゲームデザイン賞第3位　エトワールプリンセス
▶ジャンプしたときの感じや，キャラクターの質感（プヨッとした感じとか）が伝わってきて，簡単なのに飽きない。快感がある。
尾形　一朗（36）神奈川県

ゲームデザイン賞第4位　リブルラブル
▶「チェルノブ」と迷ったが，やっぱり2本のレバー（パッド）を使用する「リブルラブル」。やればやるほど頭がぷーになる。
森本　賢（20）大阪府
▶ゲームを創るということはどういうことか考えさせられる（なんちゃって）。
石田　伯仁（20）神奈川県
▶ツインレバーという特殊な操作系だが，内容はシンプルで奥が深い。10年前にこれを作ったナムコもえらいが，このゲームを移植してくれた電波新聞社もえらい。
澤田　恭幸（16）千葉県

ゲームデザイン賞選外　大航海時代Ⅱ
▶こんなタイプのゲームを私はほかに知らない！　6人の主人公，それぞれのストーリー。商人も，海賊も，冒険家もよし。このマルチシナリオがよい。　安藤　広明（20）千葉県

ゲームデザイン賞選外　ぶたさん
▶ぶたに人生ってものを教わったのれす。何事もチャレンジってね。シンプルだけど，タイムアタックが熱い。　原　真二（19）長野県

▶［自由応募部門］は38ページから始まります。

[第35回]
マルチメディアの卵

江口 響子

響子inCGわ〜るど

multi-media

multiは，「多くの」という意味。mediaは，mediumの複数形で，「手段，媒介」。直訳すると，マルチメディアは，「多くの手段」ということになります。

人間は，昔から，なにかを伝えるための手段をたくさんもっていました。言葉を話して書く，ダンスを踊る，唄を歌う，楽器を弾く，絵を描く，写真を撮る……。そして，それらを統合して映画や演劇，テレビ，絵本など，まとまったものにしてきました。

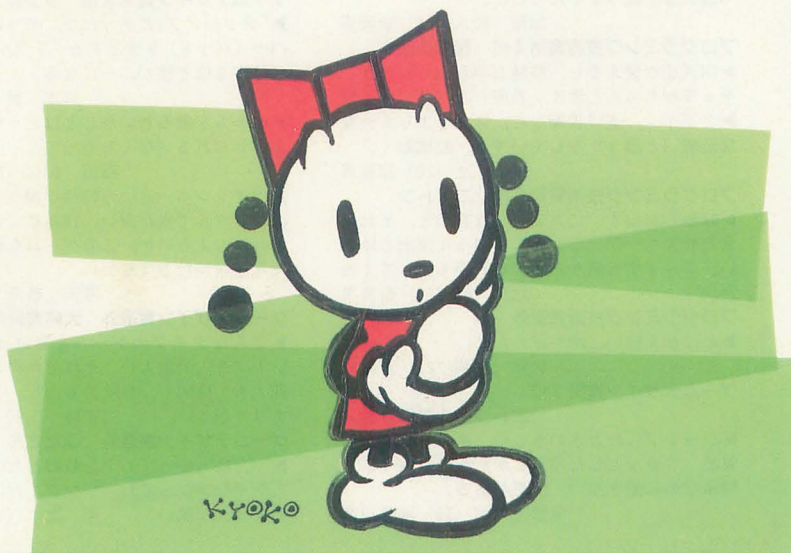
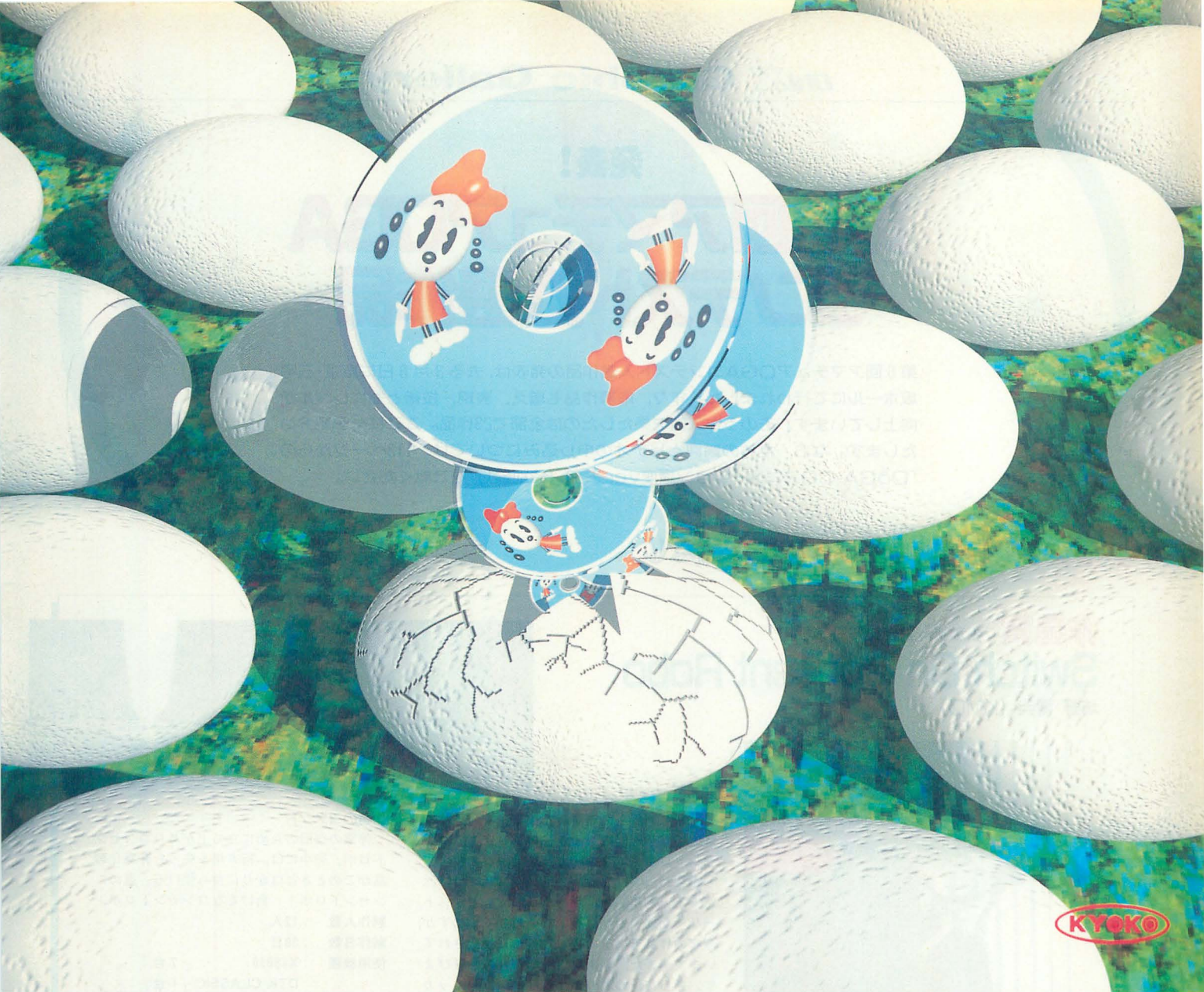
いまのところ，マルチメディアとは，画像，文字，音声が，デジタル信号に置き換えられて，ひとつにまとまったもの，ととらえられています。また，働きかけることによってゲームのように内容の変わる点が，一方的に再生されるだけのテレビや音楽CDとは異なります。

でも，マルチメディアって，本当はなんなのでしょう？

マルチメディアという言葉を聞くと，私はまっさきにCD-ROMを思い浮かべてしまいます。銀色の小さな円盤のなかに，音楽や映像，ゲームがぎっしりと詰まっている……。

最近，CD-ROM制作の話を，身近で聞くようになってきました。これは，アップル社が，新しいチップ，Power PC搭載の手頃なコンピュータを発表したこととあながち無関係ではないのかもしれませんね。Power PCは，従来の68000系のCPUを格段にうわまわる処理能力をもっています。そのため，情報量の多い画像などをスピーディに処理する必要のあるCD-ROMソフトを扱うのには向いています。

アップル社のMacintoshが，マルチメディアの最終的なプラットフォームになるかどうかは疑問の多いところですが，少なくとも，ひとつの形を示しているといえるでしょう。

CD-ROMをめぐるこうした動きは，かつてのCG制作とよく似ています。CGも以前は，スーパーコンピュータを扱える一部の人たちだけの高級なものでした。しかし，コンピュータが低価格，高性能になるにしたがって，CGはどんどん大衆化しています。皆さんがもっているX68000でも，

DōGA CGAシステムなどで，自宅でCGアニメーションを作ることができるようになっていますね。趣味でCD-ROMソフトを作るというのも，そう遠くないかもしれません。

いまのところ，CD-ROM制作のほとんどは，ビデオ，音楽CDとゲームの延長線上にあるものです。あらゆる表現手段をリミックスし，1枚のCD-ROMに収めることに，エネルギーが注がれているようです。ミュージシャン，CGデザイナ，プログラマ，ライター，ビデオエンジニアなどなど，いろいろなジャンルの人材がいままでの枠を超えて，CD-ROM制作に集まっています。

しかし，CD-ROMは，マルチメディアの卵のようなもので，まだ閉じている状態です。

マルチメディアは，リアルタイムで書き換えのできるデジタル信号の集まりです。光ファイバーによるネットワークシステムが，いまの電話のように普及する21世紀。より開かれた通信網が構築され，多くの人が同時に情報を共有したときに，卵の殻が割れてさまざまなソフトに成長するのでしょう。映像も音も，常に書き換えられて変化していくのが本質のソフトって，どんなものになるのでしょうか。気の早い話ですが，いまから楽しみにしているのです。

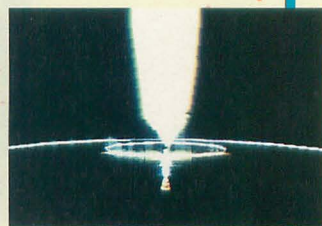
今回のCGデータ

1920×1536ピクセル
1670万色フルカラーを4×5ポジで出力
総物体数　79　すべて2次元曲面を集合演算
平行光線　1
使用ソフトは，C-TRACE
マッピングデータ作成には，C-TRACE，Z's STAFF PRO-68KとMATIER
　芝生はスキャナで取り込んだ512×512ピクセルの画像をテクスチャおよびバンプマッピングとして併用

発表！
第6回アマチュアCGAコンテスト入選作品

第６回アマチュアCGAコンテスト入選作品の発表は，去る３月６日に東京・三宅坂ホールにて行われました。年々，応募作品も増え，表現，技術ともにレベルが向上しています。そのうち入賞を果たしたのは全部で23作品。ここに一挙紹介いたします。なお，全体の講評とビデオの申し込みについては，113ページからの「DōGA CGアニメーション講座ver.2.50（特別編）」をご覧ください。

グランプリ　表彰状，賞金20万円　該当作品なし

作品賞　表彰状，賞金10万円
Switch On Concent Robo
坊野 博典（KMC）

平和な部屋を突然襲った停電。このままでは大切な予約録画ができない！　立て，コンセントロボ！　ブレーカーを上げるのだ！

相変わらずKMC(京大マイコンクラブ)は元気です。例年，オチがわかりにくいとかストーリーが弱いとかいろいろいわれていますが，今年の作品ではその辺もしっかり強化されています。また，切実なテーマが共感を呼びます。さらに，わけのわからない敵（チャッカマンなど）も笑えます。一家に一台，コンセントロボ！

作者のコメント

停電の復旧のために立ち上がったコンセントロボ。途中には，行き場を失った非電化製品がこのときとばかりに待ち受ける。進めコンセントロボ！　負けるなコンセントロボ！

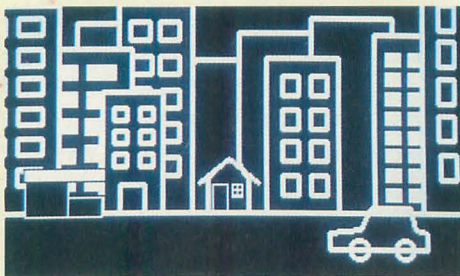
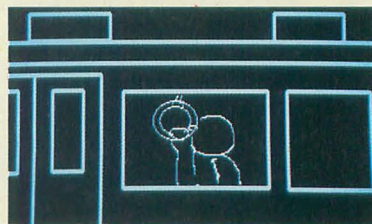
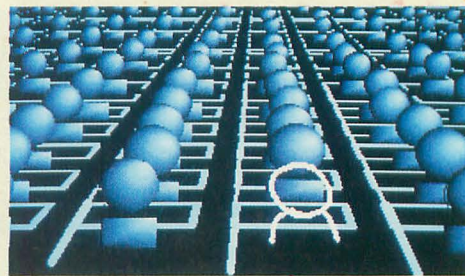
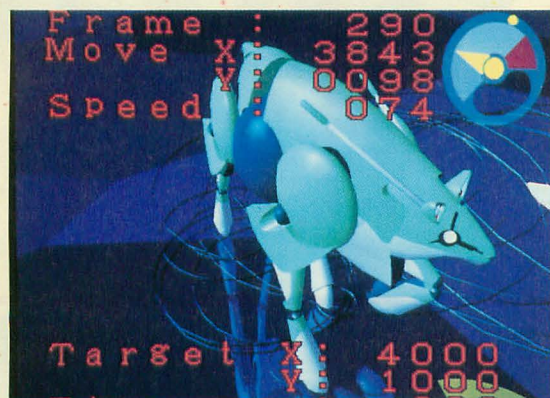
制作人数　：12人
制作日数　：50日
使用機種　：X68030　2台
　DTK CLASSIC　1台
　IBM PC互換機　1台
使用ソフト：DōGA CGAシステム

アニメーション賞
表彰状，賞金10万円
A.B.C.Day
布山 毅

２年連続アニメーション賞を受賞。昨年の「Complex」と同様，線画アニメーションで，コンセプトも表現手法もきわめてスタンダードでありながら，絶妙な演出が光る傑作です。音楽も，画面にぴったり。

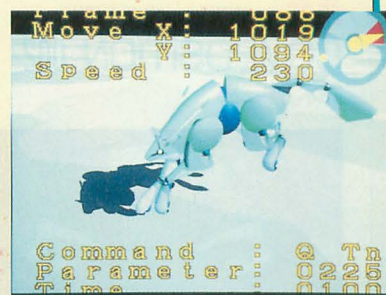
ある平凡な日常生活の異常性を鋭く表現しています。

作者のコメント

夏に１週間ほどかけて作った作品。「Complex」のような線画アニメーションをあれから何本か作ったのですが，そのうちでは結構好きな作品です。音楽は友人の富岡が作ったものを提供してもらいました。

制作人数　：１人
制作日数　：６日
使用機種　：AMIGA1200　1台
使用ソフト：Deluxe Paint Ⅳ　(AGA)
　Scenary Animator ver4.0

技術賞 表彰状，賞金10万円
HOUND'93
下岡 正道

昨年，1カット部門で絶賛された「OBJECT: MECHANICAL HOUND」がリメイクされて，昨年同様の驚異的な動きで技術賞を射止めました。きわめて高度なフレームソースを記述することで，たとえば移動速度を設定すると，自動的に歩き方を変更します。この作品では，それらの技術的なアプローチの成果をプレゼンテーション風にまとめています。とにかくカッコいい。

作者のコメント

かっこいい作品を作るという野望はあったのですが，1年が経過して手元には「犬」しかなく，結局昨年のリメイクになってしまいました。映画のコンテストに俳優だけを出すようなもので場違いな気もしますが，これが私のCGA作品です。

使用機種　：X68030 Compact
　　　　　　X68000 XVI
使用ソフト：DōGA CGAシステム
　　　　　　MATIER
　　　　　　多くの素晴らしいフリーウェア

エンターテイメント賞
表彰状，賞金10万円
冥皇龍ペルギウス
腰原 仁志

今回はやたらと宇宙バトルものが多く，審査上不利になりがちななかでの受賞です。オリジナルなデザイン，派手なバトルシーン，短い時間にうまくまとめた起伏のあるストーリー，音楽もばっちりと，ほとんどパーフェクトといえるでしょう。さらに，ヒロインも感情豊かに描かれ，新人とは思えない素晴らしい完成度です。

作者のコメント

この作品が完成したのは，CGAマガジンのおかげでした。3ヵ月ごとに締め切りがあることで，目標ができたのは大きかったです。この作品は，テロップもセリフも表情も使わずに，ただ「動き」だけで人間の感情を表現することを目的に作りました。だから変形する機動空母などは，おまけにすぎません。時間はそちらのほうに多く取られましたけど。

制作人数　：1人
制作日数　：70日
使用機種　：X68030/X68000
使用ソフト：DōGA CGAシステム
　　　　　　Z's STAFF Ver.3
　　　　　　MATIER Ver.2

審査員の紹介

森　啓次郎

パソコン専門誌「Asahiパソコン」編集長。1971年に朝日新聞社に入社。「科学朝日」「週刊朝日」の編集を経て，1991年7月より現職を担当している。

前田　徹

シャープ系パソコン専門誌「Oh!X編集長。ソフトバンクにてシャープ系パソコン専門誌「Oh！MZ」編集長に引き続き現職を担当。現在，同社のWindows専門誌「The WINDOWS」編集長も務めている。

松尾　公也

ソフトバンクのMacintosh系パソコン専門誌「MacUseer」編集長。以前にはCG雑誌の編集経験もあり。写像＋創像で3Dチャレンジしようと思ったこともあるとか。「MacUser」編集部にはなぜか3Dを趣味にしている人間が多いとのこと。

太田　修

アニメ情報誌「NEW TYPE」編集長。週刊テレビ情報誌「ザ・テレビジョン」の編集に携わったのち，1990年ビデオ情報誌「ビデオでーた」編集長を経て，1991年4月より「NEW TYPE」の編集を担当。

柴田　忠男

CGアート専門誌「Super Designing」編集長。社内にMacintoshが1台もないのに，完全データ入稿のフルDTPを実践している，とエバっている。「ディジタル・イメージ」運営委員。知的所有権研究団体「日本芸術協会」委員。著書に「SFX CM大全集」（講談社X文庫）がある。

新井　創士

パソコンゲーム専門誌「LOGiN」編集長。1983年同編集部に入り，以後，ゲームソフトの取材，編集がメイン。最近では，Windows関係などパソコンソフト＆ハードの最先端を追いかけたり，3DOでアミューズメント方面を見たりと，忙しいけれども楽しい毎日。

塚田　哲也

CGデザイナー，CGイラストレーター。1987年CGプロダクションJCGLに入社。現在はフリーで活躍中。CMの制作や個展を開くなど精力的な活動をしている。「Oh!X」の表紙も担当。

古川　タク

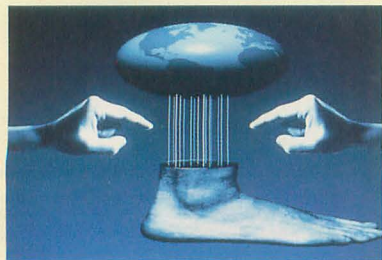
アニメーション作家，イラストレーター。1970年代後半よりコンピュータを使ったアニメーションの制作を始める。代表作は「驚盤」。

寺島　令子

ほのぼのファミリー4コマと日記漫画を得意とする漫画家。短編アニメ好きで自分でも作るのだが，割とくだらないとのこと。大阪アニメーションワークショップ林静一講座（全2日間）受講生。代表作は「くりこさんこんにちは」「墜落日誌」ほか。

鎌田　優

プロジェクトチームDōGA代表。昔々，大阪大学コンピュータクラブに入部し，CGの分科会である「prodige」を結成する。さらに，他大学のコンピュータクラブと協力して，プロジェクトチームDōGAを作る。そのままDōGAの代表にのさばり続け，現在に至る。

佳作　表彰状，賞金５万円
COMPOSITION 002
鈴木 陽二郎

独特の鈴木ワールドを展開しています。ファンの方は，パワーアップしたこの作品にはまってください。

作者のコメント

人間が無意識のうちに創り出すイメージは現実と非現実の境にある世界なのかもしれません。そのイメージを，私の頭脳の中で再構成して映像化した作品です。

制作人数　：２人
制作日数　：約200日
使用機種　：X68000　１台
使用ソフト：DōGA CGAシステム
　（CGAマガジンのデータも含む）

佳作　表彰状，賞金５万円
青年と空缶
宍戸 光太郎（あにに作人）

「猿蟹合戦」で一世を風靡し，「A PLANET」でも独自の世界を描きあげたあの宍戸さんが，禁断の世界に手を出しました。地球環境とリサイクルをテーマにした地球にやましいハードギャグ！　飛んでるカニは，13匹！　これを見ないと，宍戸ワールドは語れません。

作者のコメント

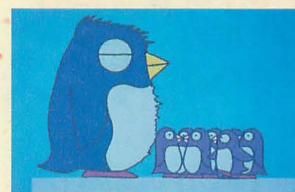
いままでの拙作では皆無であったテーマ性を追究した，新機軸の地球に優しいCGAです。作品中の空缶，飛行機のデータはLightwave3Dに添付されていたものを使用しています。

制作人数　：２人
制作日数　：20日
使用機種　：AMIGA2500　１台
　　　　　　PC-9801EX2　１台
使用ソフト：Lightwave3D ver.2.0
　　　　　　Deluxe Paint Ⅳ
　　　　　　Brillianceなど

佳作　表彰状，賞金５万円
CHUN-CHUN WORLDⅡ
佐野 元

昨年同様，かわいい鳥たちが活躍するショート・ショートの３部作です。猫との対決の続編もあります。プロのデザイナーだけあって，完成度は抜群です。でも，こうしてみると佳作に選ばれた３作は，みな昨年も活躍し，独自の路線を持っている方ばかりですね。

作者のコメント

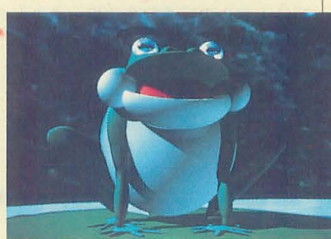
ネタ，制作ともにやっつけの感じがぬぐえないところを反省しています。ともかく，「こんなことってあるよな」って少しでも思ってもらえれば幸いです。

制作人数　：１人
制作日数　：５日
使用機種　：Macintosh Quadra800

入選　表彰状，賞金２万円
かえる
長木 功（テーマ研究VG）

いじめられ，池が嫌になったかえるは……。パーソナルリンクスを使った本格的な画面，奇妙なキャラクターなど見どころが多い作品です。

作者のコメント

かえるというキャラクターを作ってみて，かえるから見た人間の街や生き方を眺めるというような考えで作っていたのですが，時間もなくこのような形になってしまいました。なにせ初めからわからないことだらけで難しくて……。ぼくとしては，池でたくさんのかえるが鳴いてるカットは，出来がよいのではと思っています。

制作人数　：２人
制作日数　：約30日
使用機種　：PC-486
　　　　　　コマ撮り機
使用ソフト：リンクス，システム

入選　表彰状，賞金２万円
恐竜都市-1997のCM
清家 征雄

たとえ不真面目でも，いい加減でも，面白ければいいんだ！といわんばかりです。アマチュアならではのいい加減な合成が楽しい作品。

作者のコメント

アマチュアのCGでは珍しいと思われる生物を表現。しかも，これは本当に珍しいであろう実写との合成。制作意図は，単に目立つこと。内容もふざけている。

制作人数　：１人
制作日数　：50日
使用機種　：X68000
使用ソフト：DōGA CGAシステム

入選　表彰状，賞金２万円
KODOKU
中村 雄一

見知らぬベッドで目覚めた男は，不気味な物体に襲われる……。独自の質感，雰囲気を醸し出すSF大作。計算されたカメラワークも見ものです。

作者のコメント

作品の制作段階で強く感じる孤独感を，自分の視点から映像化したいと考え，制作しました。特に変わった表現方法は使っていませんが，きちんとしたCG作品を作るのはこれが初めてなので，最初の計画案が無謀だったとわかったときは，とてもたいへんでした。長い制作日数を費やしましたが，これに懲りず，以後も作品制作を続けていきたいと思います。

制作人数　：１人
制作日数　：300日
使用機種　：X68000 EXPERT　２台
使用ソフト：DōGA CGAシステム

入選　表彰状，賞金２万円

展覧会オープニングビデオ
由水 桂

展覧会の記録ビデオのオープニングタイトルだけですが，なかなかセンスよくまとまっています。

作者のコメント

頼まれて作った展覧会の記録ビデオのオープニングです。コンテを描いたときは，楽勝だ〜と思ったけど，結局４日もかかってしまった……。限られた時間内でやれることをやりました。次回こそは，大作を作りたいです。

制作人数　：１人
制作日数　：４日
使用機種　：X68030
使用ソフト：DōGA CGAシステム
　　　　　　MATIER

入選　表彰状，賞金２万円

The J-POP
宗戸 一眞

「解像連続体」などのアート派，宗戸さんが，ミュージックビデオに挑戦。編集，CG，センス抜群！　もう完全にプロの作品です。

作者のコメント

この作品は，あるバンドのライブのオープニング用に制作したものです。CGを，単なるフライングロゴとして利用したというより，編集機材の一種という感覚で使用してみました。テロップも，単なる静止画でなく動画で行うとき，CGは有効なアイテムとなります。

制作人数　：１人
制作日数　：２週間
使用機種　：X68000 EXPERT　１台
使用ソフト：DōGA CGAシステム

入選　表彰状，賞金２万円

WING CROSS
宮崎 智記

宇宙バトルもの。「VARIABLE ATTACKER」と似ていますが，こちらは１人で制作。あちこちのカットで，センスが光ります。

作者のコメント

制作期間を短くするために，形状は凝ったものではありませんが，カット割りと，動きに気を配って，テンポのある作品にしようとがんばりました。やはり，どんな方にも楽しく観ていただきたいというのがこの作品の願望であり，またこれからの目標でもあります。

制作人数　：１人
制作日数　：63日
使用機種　：X68000 EXPERT
　　　　　　X68000 Compact
使用ソフト：DōGA CGAシステム
　　　　　　Z's STAFF
　　　　　　MATIER

入選　表彰状，賞金２万円

少女キャレット　加藤 雅敏

ある気持ちのよい朝。キャレットを襲う不気味な三人衆。キャレット，いまこそ変身だ！　でも，ちょっとエグイぞ……。

作者のコメント

初めて作ったCGアニメ作品です。ちょっとコミカルなショートストーリーを作ってみました。わかりやすい作品になるよう心がける一方，使用したソフト（Imagineなど）のさまざまな機能を試してみました。私のようなCG初心者でも，独力で作品が作れるというAMIGAのCGソフトウェアの素晴らしさには，ただひたすら感激します。

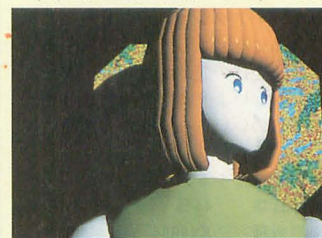
制作人数　：１人
制作日数　：60日
使用機種　：AMIGA 2000　１台
使用ソフト：Imagine 2.0
　　　　　　Deluxe Paint Ⅳ
　　　　　　Show Maker
　　　　　　Bars&Pipes Pro 2
　　　　　　Sound Master

入選　表彰状，賞金２万円

Gunner 05
小島 禎樹（Studio Dream Field）

「ある夜の出来事」の小島さんがSF大作に挑戦。ヒロイン，ティルナの表情豊かなキャラクターがいいですね。

作者のコメント

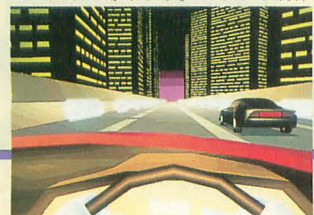
今年はほのぼの路線でなく，宇宙ものです（公約どおり？）。

制作人数　：４人
制作日数　：60日
使用機種　：Macintosh Quadra700
　　　　　　Macintosh Ⅱvx
使用ソフト：SWIREL 3D PRO
　　　　　　INFINI-D
　　　　　　Photoshop
　　　　　　Premire

入選　表彰状，賞金２万円

HELL DRIVE　白波瀬 登（白波瀬）

キャノンボールをCGA化した。いろんな車種のモデリングが見どころ。凝ったオープニング，レースの演出など，なかなかの力作です。

作者のコメント

途中に出てくる赤い車に乗った人の顔は制作者（白波瀬 登）の顔をマッピングしてあります。ですから制作者はだいたいこんな顔をしています。

制作人数　：１人
制作日数　：80日
使用機種　：X68000 XVI　１台
使用ソフト：DōGA CGAシステム
　　　　　　MATIER

入選　表彰状，賞金２万円

Fractal Atmosphere
布山 毅

アニメーション賞を受賞した布山さんの実験CGAです。うっ，酔いそう。

作者のコメント

コンピュータネットワークを映像で表現してみました。キャンパスの端末をビデオで撮影し，それをAMIGAに取り込み，いじって作りました。実験アニメーションにも似た作品がありますが，ここまで短期間で気軽には作れないでしょうね。CGAは，映画というよりアニメーションの延長と考えて，フレーム単位でいじるのが面白いと感じました。

制作人数　：１人
制作日数　：４日
使用機種　：AMIGA 1200　１台
使用ソフト：Deluxe Paint Ⅳ　(AGA)
　　　　　　VIDI AMIGA 12

入選　表彰状，賞金２万円

エンゲージ ランデブー

圡田 康司

芸術祭オープニングを作った圡田さんが，ある人の結婚式のために制作した作品です。さて，いったい誰の結婚式なんでしょう。わからないなぁ。

作者のコメント

この作品がエントリーされていると知ったのは，２月６日，もう審査も終了したあとでした。確かにDōGAに送ったけど，コンテストにエントリーするほどの作品ではないと思っていたんですけどねぇ。別にいいですけど。ああ，びっくりした。

制作人数　：１人
制作日数　：10日
使用機種　：X68030　１台
使用ソフト：DōGA CGAシステム

入選　表彰状，賞金２万円

COLOR　立石 武史

「不思議の国のアリス」をモチーフにした涙なしには見られない作品。何が涙なのかは，ビデオの解説を見てくださいね。

作者のコメント

いままで，CGA作品を，納得いく形で完成させたことがなかったので，今回は稀にみる快挙かもしれません。制作するにあたり，さまざまな困難に直面し，最後にはデータがすべて消滅してしまいました。ですから，入選してビデオとして残ったのは，たいへん励みになります。

これからも精進していきます。

制作人数　：１人
使用機種　：PC-9801XA
使用ソフト：モノリス

入選　表彰状，賞金２万円

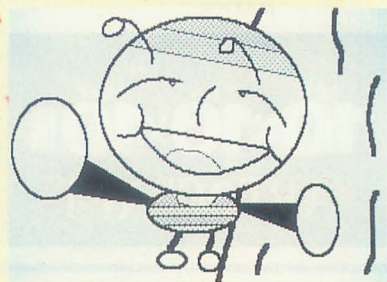
妖精BOONの大冒険

太洞 信也

村の火が消えてしまった。誰かあの山まで取りに行く者はおらぬのか！　パタパタアニメツクールでこんな大作を作るとは……。

作者のコメント

初めてパソコンでアニメーションを作ってみましたが　思った通りの動きができず時間がかかりましたが，予定していたストーリーでは時間が足りず，半分くらいになりました，が，いかがでしょうか。

制作人数　：１人
制作日数　：30日
使用機種　：PC-9821 Sl　１台
使用ソフト：パタパタアニメツクール

入選　表彰状，賞金２万円

VARIABLE ATTACKER

筑摩 昌則（大阪工業大学GR）

宇宙バトル。凝ったマッピングなど力が入っています。

作者のコメント

この作品は，とにかくかっこいいロボットアクションが作りたいと思い始めました。特にストーリーもない作品になりましたが，シューティングゲームのような感じを目指したので，全体の流れに注意し，また，カットもなるべく同じような見え方にならないように気をつけました。特に後半の基地内部のマッピング，気持ちのいいレーダーなど画面の美しさがウリです。

制作人数　：５人
制作日数　：45日
使用機種　：X68000　５台
　　　　　　PC-9801 VX　１台
使用ソフト：DōGA CGAシステム
　　　　　　MATIER
　　　　　　自作ツール

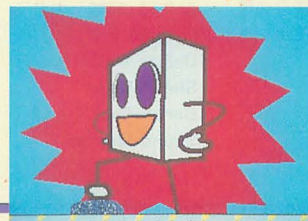
入選　表彰状，賞金２万円

THE STORY OF SOAP

立岩 潤三

『こんなソフトいらない！』の立岩さんが，「こんな洗剤いらない！」とばかりに作った社会派作品。

作者のコメント

以前に読んだ三一新書『だから，せっけんを使う』にショックを受け，これはもっと多くの人に知ってもらわなくてはと思い，この作品を作りました。合成界面活性剤の怖さが少しでも伝わればラッキーです。なお作品中，必要以上にオーバーな表現があることをお詫びしておきます。

制作人数　：１人
制作日数　：40日
使用機種　：AMIGA 2000　11台
使用ソフト：Deluxe Paint Ⅳ
　　　　　　Cinemorph
　　　　　　Vidi AMIGA 12
　　　　　　Scene Generator

入選　表彰状，賞金２万円

XEBEC　山崎 勇（JELL's TAIL）

入選した宇宙バトルもの３作品のなかでは，いちばんスピード感があります。モデリングのセンスも抜群です。

作者のコメント

この作品のあるオブジェクトは，とあるものに非常に似ています。それは認めますが，似せたわけでも，真似したわけでもないつもりです。

制作人数　：１人
制作日数　：約70日
使用機種　：X68000 XVI　１台
使用ソフト：DōGA CGAシステム

参考入選

BLUE RUNNER

圡田 康司

サンダーホーク

文月 涼

入選作品の発表会は大阪でも開催されます。
日時：1994年４月２日（土）　PM1：00〜PM4：00
場所：摂津市民文化会館
　JR京都線「千里丘」下車　南へ徒歩15分　入場無料

●紹介されたデータのうち，使用機器や使用ソフトは主なもののみで，周辺機器などは省略しています。

SOFTWARE INFORMATION

1993年のGAME OF THE YEARは決定しましたが，次回の賞候補になりそうな作品もいろいろ発表されています。1年溜めた必殺技の次は速攻？　カプコンの新作「大魔界村」は来月号でレビュー予定です。

大魔界村

大人気の「ストリートファイターⅡダッシュ」に続くカプコン4作目が早くも発売決定となった。アーケードでも好評だった「大魔界村」だ。

主人公は，勇猛果敢な騎士アーサー。3年前に大魔王の手から愛するプリンセスを救い出し，世界を守った，その人である。

あれから3年，このまま続くかと思われた平和はしかし，突然破られた。再び地上に姿を現した悪魔たちはプリンセスの城に襲いかかる。旅の空から駆けつけるアーサー。しかし，時すでに遅く，彼女は……。

災いの元凶の大魔王を倒さなければ，世の平和は戻ってこない。父親も魔族の手で失われているアーサーの憎しみ，怒りはさらに烈しく燃えたぎる。魔界に旅立ち，平和を勝ちとらんとするアーサー。行く手を阻み，次々と襲いかかる怪物たち。さあ，アクションを駆使して，戦い抜くのだ！

4月22日発売予定。CPSファイターに対応している。

X68000用　5″2HD版　9,800円(税別)
カプコン

メーカーのブランドイメージが定着？

順位	タイトル	(前回順位)
1．	ジオグラフシール	－
2．	餓狼伝説スペシャル	－
3．	EG Word	6
4．	ぷよぷよ	6
5．	龍虎の拳	－
	魔法大作戦	－
7．	エキサイティングアワー/出世大相撲	－
8．	SX-WINDOW開発キット	5
	スタークルーザーⅡ	4
10．	スーパーリアル麻雀PⅣ	－

3月号のアンケートはがきによる「期待の新作ソフト」です。このコーナーは2カ月続けてお休みだったので，前回とはずいぶん状況が変わってしまいました。3月号の発売日にはすでに前回(1月号)のトップ10中4作，特にトップの3作品が発売ずみだったので，今月は初登場が6作といつもより多めになっています。

2月号のはがきには，このアンケート項目がなかったのですが，1位の「ジオグラフシール」については欄外に「期待している」との書き込みがいくつかあったほどで，そんなところにもユーザーの熱意が見えてきます。ちなみに，獲得ポイント数もダントツです。

2位と5位には魔法株式会社の2本がやはり初登場。発売はまだちょっと先になりますが，前作「餓狼伝説2」の評価も高く，X68000ゲームタイトルが減少しつつあるなかで元気なメーカーとして期待が集まっています。

今月挙がったタイトルのうち4本は，この4月号よりも先に皆さんの手に届く予定です。来月も引き続き大きな順位変動があることでしょう。さあ，次に心躍らせてくれるのは何かな？

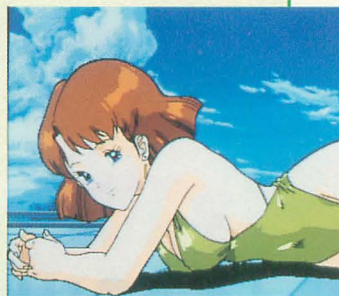
スーパーリアル麻雀PⅣ

前作「スーパーリアル麻雀PⅡ&PⅢ」が好評だったビングより，早くも次回作「スーパーリアル麻雀PⅣ」が発売される。

さて，登場する女の子は前回と同じく3人。今回は，その3人姉妹を紹介しておこう。まずはキャピキャピタイプの愛菜。15歳，蠍座の O型だ。身長は155cm，体重は45kg，スリーサイズは……(自分で画面を見て判断してくれ)。2人目はボーイッシュな悠。17歳，魚座のA型，160cmの47kgだ。最後は大人っぽい香織。19歳，双子座のAB型，159cmの48kg。写真を見れば，誰がどれかわかるだろう。

肝心の麻雀のほうは手ごわいぞ。3人の思考ルーチンはアーケード版を再現とのこと。今度こそ仇が取れる。「え〜，難しいのはチョット」という人でも大丈夫，難易度の設定もできるのだ。もちろんクリアしたところまでのアニメーションを再生することもできるぞ。ほかにも，パソコン版オリジナルの「麻雀バトルモード」を搭載している。

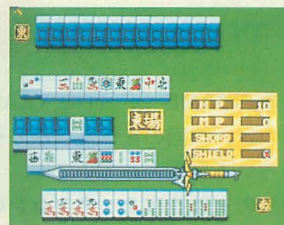
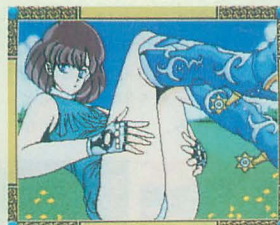
発売予定は4月中旬，もうすぐだ。

X68000用　3.5/5″2HD版　12,800円(税別)
ビング　☎03(5496)2501

マージャンクエスト

RPGプラス麻雀，その名も「マージャンクエスト」。ほとんど完成間近のようである。

これまでに紹介したノーマルモード，アドバイスモードのほかにトーナメントモードが加わっている。このモードでは，雀魔王コクシーを倒したあとの世界が舞台だ。リュウコは聖雀士となったものの，最近は麻雀が流行らなくなったおかげですっかり貧乏になってしまった。そんなリュウコのもとに1枚の紙切れが……。それは，テンパネル王国で開催される麻雀大会の知らせであった。そしてリュウコは賞金を目指して旅立っていく。

このモードは1回勝負，先に和了るか，流局時にテンパイしていたほうが勝ちである。勝てば相手のモンスター(女の子)はもちろん……。ただ，ほかのモードで使える各種の技はここでは使えない。麻雀の実力のみでの勝負。いまから腕をみがいておくのだ。

X68000用　5″2HD版　価格未定
SPS　☎0245(45)5777

アルゴスの戦士

いまから10年前にテーカンという名前のメーカーがあった。「スターフォース」や「ボンジャック」といったヒット作を連発していた3歳児でも知っている人気メーカーだった。それが突然1986年にCIしてテクモという名前になったときにリリースされたのが，この「アルゴスの戦士」である。今回，電波新聞社のビデオゲームアンソロジーの第9弾として，この「アルゴスの戦士」がX68000に登場の運びとなった。

鎖鎌のような円盤を投げ，振り回し，敵の真っ只中を駆け抜けていく。美しいグラフィックにハードなアクション。テンポもよく，ほどよい隠し設定やテクニカルなボーナスの設定などの奥の深さは，いまのジャンプアクションの基礎であり，お手本のようでもある。とにかく攻撃・移動・ジャンプのすべてが気持ちよく，かつ楽しいのだ。

格闘ばかりがゲームではない。ゲームのジャンルが数多くあり，それぞれのジャンルに名作があることを，この「アルゴスの戦士」は教えてくれるだろう。

4月下旬に発売予定。　(八)

X68000用　5″2HD版　5,300円(税別)
電波新聞社　☎03(3445)6111

計測技研
フリーソフトウェアの募集

計測技研は，1992年に発売したX68000用CD-ROM「フリーソフトウェアセレクション」の第2弾として，X68000ユーザーの作った各種プログラム，音楽，画像，文書などを収録したCD-ROMの発売を予定している(価格は未定)。そのための各種データなどを募集している。

応募期間は4月末日まで。応募要項などの問い合わせは，パソコン通信にてもできる。

問い合わせ先

ネット

NIFTY-Serve TAB02267　Network SX-68k #37
SUNDAY-NET sun3571　SPS-NET 13
Pekin-NET BASIC_H　梁山泊 BHOUSE
TECOSYS-2 BASIC_H

住所

〒320　栃木県宇都宮市京町11-18
OYAMAビル2F
(株)計測技研 技術開発部
フリーソフトウェアセレクション2担当
☎0286(38)0301　FAX0286(38)0305

デジタルアートコレクション vol.9

パソコン通信をやっている人でなければ手に入りにくいアマチュアCG作品を集めたデータ集のシリーズ。3月号でも紹介したが，さっそく続編のvol.9が発売された。ダウンロード時間を短縮するためのファイルサイズ縮小化を考えなくていいため，パソコン通信で配布されているものに加筆，修正を加えた作品もある。

vol.9は4096色中16色の作品集でPC-9801，FM TOWNSでの再生も可能。

通信販売も行っているので，希望する場合は，CONNECTLINEまで電話にてお問い合わせを。

X68000用　3.5/5"2HD版　1,500円(税別)
ブラザー工業(TAKERU)　1,200円(税込)
CONNECTLINE　☎0899(26)7821
ブラザー工業(TAKERU)　☎052(824)2493

宝魔ハンターライム9

第7話からディスク3枚組となった「宝魔ハンターライム」シリーズ。アニメーションもずいぶん増えてパワーアップされている。

さて，第9話では新しいキャラクターが登場する。突然バースを訪ねてきたのはなんと，可愛い女の子。しかも，連れているのはプギに似た……？　彼女の名前はココナ。そう，ライムたちと同じ魔界の女の子だ。はたしてココナの正体は？　そして，ココナとバースの驚くべき関係とは？　ライムとバースの仲はどうなる？

4月10日発売予定。

X68000用　3.5/5"2HD版　1,500円(税込)
ブラザー工業(TAKERU)　☎052(824)2493

画面はPC-9801版です

発売中のソフト

★EGWord SX-68K　シャープ
X68000用　3.5/5"2HD版　59,800円(税別)
★SX-WINDOW開発キットWorkroom SX-68K　シャープ
X68000用　3.5/5"2HD版　39,800円(税別)
★SX-WINDOW開発キット用サポートツール集　シャープ
X68000用　3.5/5"2HD版　12,800円(税別)
★Hyper Pixel Worksエクステンション1　マルチフォントシステム　ブラザー工業(TAKERU)
X68000用　3.5/5"2HD版　1,500円(税込)
★Hyper Pixel Worksエクステンション2　カラーコントロールセット1　ブラザー工業(TAKERU)
X68000用　3.5/5"2HD版　1,500円(税込)
★デジタルアートコレクション vol.9　CONNECTLINE
X68000用　3.5/5"2HD版　1,500円(税別)
ブラザー工業(TAKERU)　1,200円(税込)
★エキサイティングアワー/出世大相撲　電波新聞社
X68000用　5"2HD版　5,300円(税別)
★宝魔ハンターライム8　ブラザー工業(TAKERU)　3/10
X68000用　3.5/5"2HD版　1,500円(税込)
★ジオグラフシール　エグザクト　3/12
X68000用　5"2HD版　9,800円(税別)

新作情報

★レッスルエンジェルス2　ブラザー工業(TAKERU)　3/19
X68000用　3.5/5"2HD版　4,900円(税込)
★ぷよぷよ　SPS　3/25
X68000用　5"2HD版　8,800円(税別)
★宝魔ハンターライム9　ブラザー工業(TAKERU)　4/10
X68000用　3.5/5"2HD版　1,500円(税込)
★麻雀航海記　ブラザー工業(TAKERU)　4/未
X68000用　3.5/5"2HD版　5,800円(税込)
★マージャンクエスト　SPS
X68000用　5"2HD版　価格未定
★宝魔ハンターライム10　ブラザー工業(TAKERU)　5/10
X68000用　3.5/5"2HD版　1,500円(税込)
★ロボスポーツ　イマジニア
X68000用　5"2HD版　価格未定
★Traum　象スタジオ
X68000用　5"2HD版　価格未定
★鮫！ 鮫！ 鮫！　KANEKO
X68000用　5"2HD版　価格未定
★達人　KANEKO
X68000用　5"2HD版　価格未定
★エアバスター　KANEKO
X68000用　5"2HD版　価格未定
★サバッシュII　ポプコムソフト/グローディア
X68000用　5"2HD版　価格未定
★麻雀悟空・天竺への道　シャノアール
X68000用　5"2HD版　9,800円(税別)
★スタークルーザーII　アルシスソフトウェア
X68000用　5"2HD版　価格未定
★魔法大作戦　EAビクター
X68000用　5"2HD版　価格未定
★あすか120%　ファミリーソフト
X68000用　5"2HD版　価格未定
★スーパーリアル麻雀PIV　ビング　4/27
X68000用　3.5/5"2HD版　12,800円(税別)
★龍虎の拳　魔法株式会社
X68000用　5"2HD版　価格未定
★餓狼伝説SPECIAL　魔法株式会社　7/未
X68000用　5"2HD版　価格未定
★アルゴスの戦士　電波新聞社　4/28
X68000用　5"2HD版　5,300円(税別)
★大魔界村　カプコン　4/22
X68000用　5"2HD版　9,800円(税別)
★地球防衛MIRACLE FORCE　カスタム
X68000用　5"2HD版　価格未定
★Mu-1 GS　サンワード　4/中
X68000用　5"2HD版　価格未定

飛んで，撃て！ 跳んで，踏め！

Sudo Yoshimasa
須藤 芳政

あの「まる文字」姫リルルちゃんで僕たちのココロを虜にしたまま沈黙を保っていたエグザクト。月日は流れ，1年ぶりの嬉しい再会です。今度はポリゴンだあっ！　ということで，またまたX68000ユーザーゴコロをくすぐります。

エグザクトの新作は何だろう？　と，首を長くして待った末，めでたく首の関節を1個増やすことに成功してしまって女の子にモテモテなあなたへ。3Dポリゴンシューティングゲーム「ジオグラフシール」の登場です。ひょっとしたらこれを読んでいるときにはもうすでにプレイしているかもしれませんね。

現在手元にあるサンプル版では3面の途中までしか遊べないのが残念ですが，グルグル動きまくるポリゴンは迫力十分。アクションゲーム好きのX68000ユーザー犬がいたら，きっとヨダレたらしっぱなしになっちゃうソフトです。

まずは操縦に慣れよ！

ゲムオ「こんにちはー」

よ「やあ，ゲムオ君。これを見たまえ！」

ゲムオ「あっ，これはエグザクトの『ジオグラフシール』じゃないですか！　確か，AIR PERSPECTIVEとかいうのがウリの……。で，AIR PERSPECTIVEっていったい何なんですか？」

よ「さすがゲムオ君。目の付けどころがシャープじゃないか。AIR PERSPECTIVEというのはズバリ『空間での自由な行動が可能』ということだ」

ゲムオ「それって，3月号のエグザクトの広告でAIR PERSPECTIVEの下に書いてるけど本当は空気遠近法って意味じゃ……」

よ「おんどりゃー！」

ポカ！

よ「我々一般人はそんな難しいことは知る必要なしだ！　つべこべいわずに男は黙ってレッツプレイ！」

敵ながらカッコイイねェ，いよ男前！

ゲムオ「イテテ，それじゃあぼくはジョイパッドでやってみます。自機は左右のパッド操作で左右に旋回，前と後ろで前進と後退ですね。トリガを押すと弾を発射。簡単だーい！　わわ，でも自機の移動が遅くてどんどん敵の弾くらっちゃうよ」

よ「ふふ，もうひとつのトリガを押してみるがよい」

ゲムオ「おお，ジャンプした。これなら敵の弾も簡単にかわせるね」

よ「さらに，ジャンプ中にもう一度ボタンを押すことにより4段階まで連続して上昇できるんだ。貸してみなさい，ほら，こんな風に……」

ゲムオ「ひゃー，敵があんなに小さく見える！　でも，なんで2段階以降のジャンプは目線が地面と垂直になるのかな？　それにこんなに上昇しなくたっていいと思うんだけど……」

よ「ふふ，たとえば，高いところからものが落ちてきたらどうなると思う？　隙あり，ドロップキーック！」

ばきゃ！

ゲムオ「ひぃぃぃ。痛いっすよー！」

よ「だろ？　自機は上昇して敵の上に降下することによって相手にダメージを与えられるんだよ。高く上昇すればそれだけ効果も大。目線が地面と垂直になるのは降下位置を合わせやすくするためだ」

ゲムオ「そっかー。よーし，ジャーンプ！　あたたたたたたた……おあたー！」

よ「バカヤロー！」

ドスッ！

ゲムオ「う！　何するんですかー！」

よ「やみくもに連打してもだめだ！　タイミングよく目押しするのだ」

ゲムオ「おお，できたできた。これでぼくもいっぱしのジオグラプレイヤーだね！」

よ「こ～いつぅ～」

2人「あーっはっはっは」

1面：砂が舞っているから鳥取？

よ「さて，ゲムオ君。調子はどうだね？」

ゲムオ「いくら敵をやっつけてもボスが出てこないんですけどー」

よ「それは条件を満たしていないからだよ。ステージのあちこちに『TARGET』ってマーキングされた敵がいるだろう？」

ゲムオ「うん，いたいた」

よ「その敵を倒すごとに右上に表示されている『TARGET』の数が減っていって，それが0になって初めてボスエリアへの入り口が開くんだよ」

ゲムオ「じゃあ，さっき通行止めになってたところがボスエリアの入り口なんだね」

よ「そう，すべてのターゲットを倒したと

地図があれば犬のおまわりさんは無用だ

X68000用　5"2HD版　9,800円（税別）
エグザクト　☎025(284)7304

きがハンターチャンス（by柳生）！　なるべくならシールド満タンでゴールデンハンマー（平日午前中に暇な人ネタ）！」
ゲムオ「おーい，やまだくーん！　座布団1枚持ってっちゃって」
よ「たわけー！」
ぼぐ！
ゲムオ「うへえ！」
よ「ゲームに集中せんか！　見ろ，ボスへの扉が開かれたぞ！」
ゲムオ「兄貴！　おいらいま，さいっこーに燃えてるっすよー」
メラメラ……

1面のボスは燈台だ

ゲムオ「あ，ボスの周りをアイテムが回ってますよー！　いただきまーす！」
よ「とち狂うんじゃない！　あれはボスをガードしているボールだ。まずはあのボールを破壊しろ」
ゲムオ「ジャブだ，フックだ，ストレートだ！　あーあ，もうやられちゃったよー」
よ「若い読者が理解できないネタはよせ。あのボールは急降下の踏みつけで破壊するのだ。まあ，別の方法がないわけでもない。ボスまでも倒してしまうという……」
ゲムオ「ええ!?　それってどういう方法ですか？」
よ「ヒントは『グルグル回る』だ。だからといって君自身が回るようなボケはしないように」
ゲムオ「へっへ，だんな。今日は先読みが冴えてるじゃないスか。わかったぞ！　ほれ，武器をホーミングに換えてグルグルー。なあんだ，これならダイヤルQ²に電話しながらでもボスが倒せるじゃないですかぁ」
よ「とかいいながら私の電話を使うんじゃない！」

2面：ベイブリッジがあるから横浜？

よ「次は2面だけど，もう余裕だよね」
ゲムオ「ふぅ，ぼくの手にかかればどんなゲームもワン・ドッグですよ」

1面のボス殿はもう爆発確定だ

よ「イチ・コロといいたいんだね。でもいまどきコロなんて名前の犬いるのかな？　まあ，いいや。えい！」
ゲムオ「あれえ？　いままで敵にくっついてたマーキングが消えちゃいましたよ。だんなー，ふざけてもらっちゃ困りやすぜ。これじゃ，どれがザコでどれがターゲットなのかわからないじゃないですか」
よ「だから面白いんじゃないの」
ゲムオ「おんどりゃー！」

2面のボスは札幌かに○○駅前本店だ

よ「あれえ？　ゲムオ君，まだボスと対決できないの？」
ゲムオ「どこ行ってたんですかー。いまやっとボスのところまできたんですよ。ほらほら，ぼくの華麗なパッドさばきを見ていてください」
よ「ふっふ，ゲムオよ，世の中そんなに甘(あま)うないで」
ゲムオ「うわ！　カニだかゴキブリだかわからないやつが出てきたぞ！　何だこのミサイルはー!?　なーんてジャンプすればけっこうかわせるじゃないですか，ふーんふふーん（by志茂田○樹）」
よ「ゲムオくーん。今日，君ん家の晩ご飯なぁに？」
ゲムオ「いま集中してるんですから，話しかけないでくださいよ！　あれ!?　あの長いレーザーみたいなのは？」
ビビビビ，プチン……
よ「あー，残念だったねー。あれはジャンプで避けなきゃいけないらしいんだよ」
ゲムオ「え？　『らしい』って。さては芳政兄貴も倒せないんだな？」
よ「ゲムオ，おまえそろそろ帰るか？」

えい！　おまえなんか踏んづけてやる

これは久々に熱くなれるかもしれない

実はこのゲーム，RS-232Cケーブルを使っての対戦モードも用意されている。残念ながらX68000が2台ないと遊べないし，もちろんモニタも2台必要。遠方の友人に「うちのロクハチ壊れちゃって……。修理してる間だけちょっと貸してよー，モニタもついでにね」と言葉巧みに借りて，本人が直接取りにくるまで返さないというのを最後の手段として考えておこう。

対戦は，1つのフィールド内で各自選択したマシンで飛び回って撃ち合う。対戦数は，1，3，5戦が選べる。負けると，人間相手なので当然ながら非常に悔しい。頭上をどんどこ踏みつけられるとプライドある大人としては許せない……など，対戦だけでもかなり長く遊べるだろう。

このゲーム，ジャンプするとどうしても自分の体も一緒に動いてのけぞってしまう。ほかの人がプレイする姿を眺めてみても……うん，確かにのけぞっている。それだけ臨場感たっぷりということだ。さあ，きみも早くこの「ジオグラフシール」でのけぞるのだ。エビビ！

武器を供給してくれるなんて変な敵だ

敵がわらわら出てきたら即ジャンプ！

製品版が待ち遠しいな，わくわく

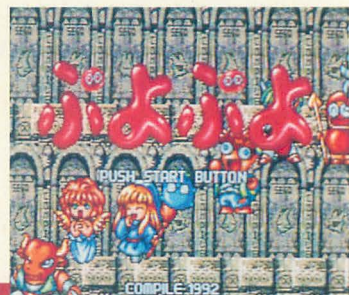
エグザクトさんは，本当に期待を裏切りませんねー。早く製品版で遊んでみたくてしょうがありません。音楽は内蔵音源のほかにGS系のMIDI音源に対応していて，RS-232C端子からでもMIDI信号を送出してくれるようです。もちろん曲は「かっちょええ」曲ばっかり。

あと，サンプル版ではできませんが，ユーザーディスクに途中のデータをセーブできるようにもなるはず。

総合評価	0　　5　　10
うごき	★★★★★★★★
音楽	★★★★★★★
のけぞり度	★★★★★★★★★

あなたのハートにだいあきゅーと!

対戦ゲームが花盛りですが，血を見るのが苦手だなんて，そんなやさしいアナタに贈る平和で熾烈な勝ち抜き戦です。ぷよぷよ落ちてねちゃねちゃくっついてぽわんと消えるけど，ほのぼのしてたら，どどーんと降ってくるのでご用心。

Takahashi Tetsushi
高橋 哲史

だだだだいあきゅーっ！　みなさま元気に連鎖してますか？　私はいつもニコニコの5連鎖を仕込んで悦に入っています。そうです，ついにX68000にも「ぷよぷよ」がやってきたのです。え,「ぷよぷよ」って何？……いけませんねぇ，落ちゲーの傑作「ぷよぷよ」をご存じないとは。そんな君は，岩石5個でおじゃまぷよの海に沈めちゃうぞ！　それが嫌なら，これを読んで一緒にぷよぷよワールドにレッツゴーだぁ！

これがぷよぷよだあっ

ぷよぷよとは何か。その正体はテトリスやコラムスと同様のいわゆる「落ちゲー」(落ちてくるモノを揃えて消すゲーム)なのです。ところがどっこい，そこらの落ちゲーとは面白さがひと味もふた味も違います。なんてったって私はコレのせいで，メガドラのパッドをひとつ潰したんですから……。

さて，ゲームは，上から2個セットで落ちてくるぷよを色別に4個以上揃えて消していきます。テトリスやコラムスなどと違って，必ずしも直線状でなくても，たとえばL字形やT字形に揃えてもぷよは消えます。この辺のフレキシビリティ(?)がゲームに幅を出しているといえるでしょう。

そしてぷよぷよの醍醐味はなんといっても，連鎖！　3連鎖4連鎖と次々にぷよが消えてゆくさまは，快感を通り越してある種の感動を伴います。この連鎖を意図的に組めるようになって初めて，真に「ぷよっている」ことになるのです。連鎖が増えるほど，相手エリアに降るおじゃまぷよの数も激増しますから，相手をいたぶる楽しみも増すというものですな。ほっほっほ。

X68000用　5″2HD版　8,800円
SPS　☎0245(45)5777

まずは練習だあっ

手始めに，3ステージで終了する練習モードに突入です。本当は対戦ぷよがいちばん燃えるんですが，何事も最初は修行が肝心。特にぷよぷよの場合，いきなりうまい人と対戦しても，何もさせてもらえないうちにおじゃまぷよの海に沈められてしまうのがオチです。まずは練習モードで効果的な積み方を体で覚えることが先決なのです。

練習モードでの最初の相手は，スケルトンT。こいつは，ぷよの回転すらできないようなバカなので，練習相手にはもってこいです。そこで図を見てください。これが「2段積み」「3段積み」による連鎖の基本形です。何も考えなくてもこの形にぷよを積み上げられるようになったらしめたもの。2段積みよりは3段積みのほうが応用範囲が広いので，できればこっちを会得しましょう。いつも思い通りのぷよが降ってくるとは限らないので，多少色違いぷよが交ざってもきちんと連鎖する3段積みのほうがお得です。まあ感覚的には2段積みのほうがわかりやすいので，そちらを先に修得してから3段積みにステップアップ，というのがいちばん確実かもしれません。

それともうひとつ知っておきたいのは，ぷよの特性です。ぷよは2個セットで降ってきますが，片方をほかのぷよにひっかけて置くことで，2つに分離して置けるのです。これがスムーズにできるようになると，いわゆる無駄ぷよが少なくなり効率的な積み方ができるようになります。

勝つための基本は，連鎖を起こしまくって，相手エリアにおじゃまぷよを大量に降らせることですから，とにかく2連鎖3連鎖4連鎖5連鎖とどんどんぷよを組むことが大切です。ちなみに8連鎖すると相手エリアはおじゃまぷよで埋まってしまいます。

スケルトンTのあとに出てくるナスグレイブとマミーも連鎖は組んできません(さすがにぷよの回転はしますけど)ので，ここで心ゆくまで練習を重ねてくださいね。

目指せ，ぷよマスターっ！

さて肩慣らしが終わったところで，いよいよ本格的に「ひとりでぷよぷよ」に挑みます。まずここでは，全13人の敵キャラクターの特徴と攻略を見ていきましょう。

1.ドラコケンタウロス

ドラコは連鎖も組んでこないし，積むスピードも遅いので練習モードをクリアした方なら簡単に勝てるでしょう。ま，練習モードの延長って感じですかね。

2.すけとうだら

積むスピードがドラコより若干速くなっていますが，本格的に連鎖を組んでくるわけではないので，それほど強敵ではありません。これも楽勝でしょう。

3.スキヤポデス

同上。それにしてもこいつの足はほんとにでかいなー。

4.ハーピー

まずは練習モードでしっかり修行

ゲームスタートと同時にいきなり左右に積み上げるという，あからさまにフィーリング連鎖狙いの戦法をとってきます。フィーリング連鎖というのは，2段積みや3段積みのように意図的にぷよを積んでいくのではなく，同じ色のぷよを3つずつ適当に積んでいって「これで連鎖するんじゃないかなー」というところで下層のぷよを消すという実にアバウトな戦法です。しかし，これが始まると，連鎖が結構続いてしまうこともあるのでなかなかあなどれません。ちなみにフィーリング連鎖を極めたい方(そんな人いるのか)はこのハーピーの積み方を参考にするとよいかもしれませんね。

5．さそりまん

このあたりになると，本格的に敵も連鎖を組んできます。ただしこいつは下手な2連鎖しか組んでこないので，それほど苦戦することはないでしょう。

6．パノッティ

ここが前半のヤマかな？　可愛い顔をしながらも鬼のような速さでぷよを消していきます。連鎖を組んでくるわけではないのですが，まばらに，しかしどんどん降ってくるおじゃまぷよがかなり目障りになることでしょう。とにかく早めに仕掛けを完成させてしまうことを心がけましょう。

7．ゾンビ

こいつはそれほど強くありません。さそりまんと同じような攻撃をしてきます。

8．ウィッチ

基本的にはパノッティと同タイプ。さらにスピードが速く，結構辛いです。ちなみにこのウィッチは，ぷよを消すときにお星様を飛ばしてくれます。おしゃれですねぇ。

9．ぞう大魔王

10．シェゾ

11．ミノタウロス

ステージ9からはBGMも背景も変わって気分を一新すると同時に，さらにきびしい攻撃が待っています。この3人は基本的にはパノッティ型ですが，とにかくぷよを積むのが速い！　「そんなんありー!?」と叫びたくなるようなスピードで積んできます。たいていの仕掛けはおじゃまぷよに埋められてしまいますから，重要なのは「埋められた仕掛けをいかに速く掘り出すか」です。絶え間ない2連鎖でおじゃまぷよを粉砕していきましょう。あるいは大きい仕掛けはあきらめて3連鎖くらいを連発する，という作戦も結構有効かもしれません。

12．ルルー

13．サタンさま

何もいうことはありません。努力と根性です。なにしろこの2人は凄まじいスピードでの積みに加えて，連鎖まで組んでくるので手のつけようがありません。100回くらい戦えば心眼も開いてくるでしょう。

ちなみに私は，コンティニューの嵐の末，約1時間ほどでぷよぷよ音頭を聴くことができました。皆様もぜひぷよマスターの称号目指して頑張ってください。ここまでくれば，対戦ぷよもそんじょそこらの人間には負けないようになっていることでしょう。

のどかなビジュアルシーンもそのままだ

よぉし，いよいよこれから5連鎖だっ！　覚悟せいっ！

8連鎖で，相手エリアはおじゃまぷよだらけ(残りの岩石に注目！)

これは本当のぷよぷよだっ

さて，気になるX68000版の出来ですが，まずまず合格点をあげても問題ないでしょう。大量のおじゃまぷよが降るときはちょっとがたつくような気もしますけど，10MHz機でも十分遊べるスピードだし，ちゃんとぷよぷよもぷよぷよしてます。もちろん，連鎖のときの掛け声も「ふぁいやーっ」から「ばよえーんっ」までばっちり入っています。面移行時のデモもプリティで，まさにアーケードそのままといった感じです。

ちょっととまどったのは，ぷよの回転が1ボタンなところ。アーケード版は確かにこうなのですが，メガドラの2ボタン操作に慣れてしまった私には，ぷよの逆回転がないのはちょっと辛かったです。ま，これは慣れの問題ですが……。あとメガドラ版にあった対戦モードのハンデ(甘口，中辛，辛口,激辛)が採用されていないのもちょっと残念。ハンデをつければ初心者対上級者でも結構いい対戦ができるんですけどね。

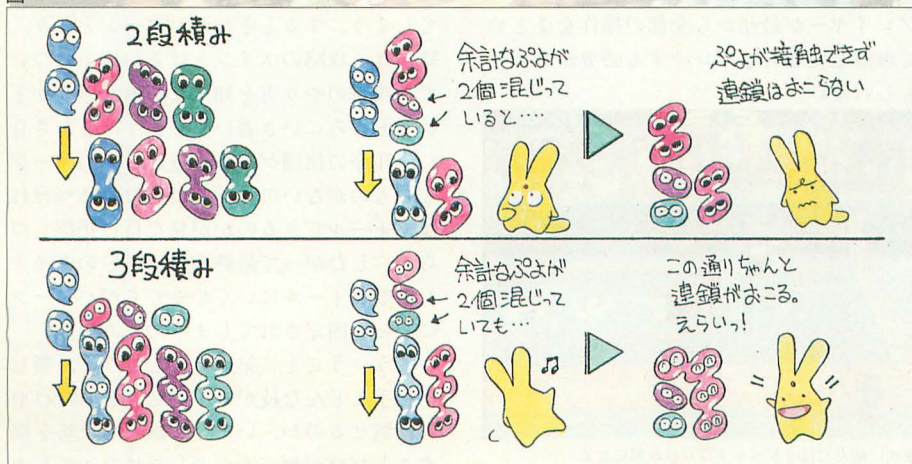
ということで，「ぷよぷよ」の魅力を余すところなくお伝えしたつもりですが，いかがでしょうか。シンプルながらバランスのとれた奥の深さに，対戦の熱さと連鎖の壮快さがほどよくあいまって，長くプレイできることうけあいです。きっとあなたの夢のなかにも，おじゃまぷよが大量に出てきてくれることでしょう。

図

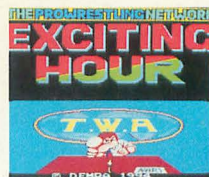
できればオマケも欲しかったな

全体的に非常にナイスな移植で，Oh!Xのぷよマスターと呼ばれる(？)私も大満足なのですが，出来がよいだけにちょっと欲も出てしまいます。というのもメガドラ版にあった「とことんぷよぷよ」(ひとりでひたすらぷよを積み続けるモード)と，ゲームギア版にあった「謎ぷよ」(ぷよの詰め将棋みたいなもの)もオマケとしてついていればなーと思ってしまうんです。そうすれば「X68000ではすべてのぷよが遊べる！」と自慢できたんですけど……やっぱり贅沢ですかね。

総合評価

	0 — 5 — 10
グラフィック	★★★★★★★★
サウンド	★★★★★★★★
操作性	★★★★★★★
小粋なカーバンクル	★★★★★★★★★

裸の男同士がもつれて嗚呼！

格闘技もさまざまあれど，現実の日本で最もメジャーなのは，喧嘩を除けばやっぱりプロレスと相撲。今回のビデオゲームアンソロジー作品は，その両方が闘える2段攻撃。チャンピオンベルトと横綱の名誉，先に狙うのはどっち？

Yaegaki Nachi
八重垣 那智

私の記憶に間違いがなければ，不幸は幸せなときにしかやってこないそうである。

ある日いつものように編集部に顔を出すと，私はこう聞かれた。「エキサイティングアワーってゲーム知ってる？」。そこで私はすかさずこう答えた。「それはブギーマナーとバトルレーンVOL.5の前に出た，ゼロイゼとバッテンオハラのスチャラカ空中戦で有名なテクノスジャパンのプロレスゲームですね」と。

ところが，得意気に高くなった鼻をさする私に向かって返ってきた言葉は，思い出すのも嫌な，恐ろしいものであった。「やっと『知ってる』人がいたよ，じゃあよろしく」。このとき私は自分が年老いつつあることを初めて実感し，それを呪った。しかし，いまどきの若い者(ゲーマー)が，このゲームを知らないとはケシカラン。本当は見たくないけど，親の顔が見てみたいものである。

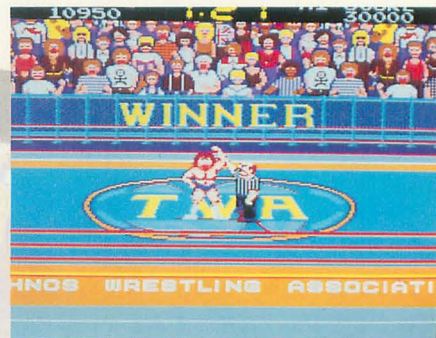
エキサイティングな絶叫

定期的に名作を発表している電波新聞社のビデオゲームアンソロジーシリーズであるが，今回はテクノスジャパンという通好みのセンで攻めてきた。「エキサイティングアワー」と「出世大相撲」の2タイトルのカップリングで発売である。冒頭のOh!X編集部におけるやりとりはかなり脚色したものだが，やはりそれでも「こんなゲームなんて，どっちも見たことも聞いたこともない」という人は少なくないだろう。確かにいままでのシリーズに比べると，人気・知名度共にランクが違うという印象は否めない。そこで，それぞれのゲームごとの内容を押さえて，ゲームを理解することに重点をおいて解説していこう。

X68000用　5"2HD版　5,300円(税別)
電波新聞社　☎03(3445)6111

まず,「エキサイティングアワー」は，1985年にリリースされたプロレスゲームである。目的は，勝ち続けてチャンピオンになること。ルールは一本勝負。一度でも負けると，即ゲームオーバーとなってしまう。継続プレイや2P同時プレイなどはなく，ひたすら確実に黙々と相手をフォールしていかなくてはいけないゲームである。

操作は，互いに組み合って，レバーの方向と大小の技ボタンの組み合わせで，プロレスの派手な技を画面に炸裂させることができる。最近の流行モノのような「PPPK」といった連続シーケンスや，コマンドと呼ばれるような技はもちろんない。またレバーの方向とボタンの組み合わせによる技の相関関係も，頭に入れておくことが必要である。どれが大技で，どれが小技で出るのかという程度は把握しておかないと，相手と組んでも思うように技をかけることができないという状態になるからだ。とりあえず簡単な技だけを覚えておくのではなく，プレイヤーが最初から全部の操作をまとめて理解したうえでプレイする必要があるかもしれない。

有利に組んだほうがヘッドロック状態になる

まずは一勝。チャンピオンへの第一歩か？

リングは戦場だ

操作が理解できたら，ゲームに入るのだが，展開としてはどの面も同じ場所で違う相手と試合をするだけの比較的単調なもので，そこで勝てばよい。しかし，これはあくまでゲームなので「どうやれば相手を倒すことができるのか？」というところにこのゲームの駆け引きが集約されている。

敵にはそれぞれ特徴があって，敵によってかかる技と，逆にやられてしまう技がある。しかも有効な攻め方となると，敵ごとにかなり限定されてしまうのである。よって，敵Aに対してはパンチで倒してロープに飛ばして大技，というような方法がわかれば，あとはそれの単調な繰り返しになってしまう。するとそれがパターンとなり，結局は，攻略のポイントはその相手について，攻撃のやり方を知っているかどうかというところにいき着いてしまう。またさらに，昨今の格闘ゲームと違い，体力ゲージなるものがないので，どれだけ痛めつければフォールできるのかが見た目に把握しづらい。したがって最終的には，どのタイミングでフォールにいくかまでもがパターンになって固定されてしまうのである。

こういうことに気づいてしまうと，新しい相手にどんな技が効くのか，どれだけやれば倒せるのか，いろいろ試して攻略を探すことだけがゲームの楽しみになってしま

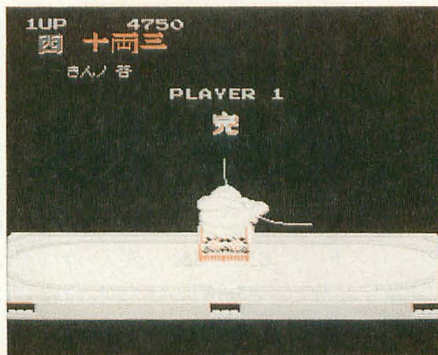
うのは否定できない。つまりはそこに魅力を感じなければ，理由のわからないまま，自分のかけているつもりの技がかからずに，一方的に敵にやられてしまうという不条理さに我慢がならないだろう。

今回のX68000版をプレイしてみて，移植という観点で見れば，こういった技のシステムや操作感などは合格レベルだと思われるが，9年前のゲームだという事実は事実であると認識しておいたほうがいいだろう。また，細かいことをいうと本物は縦画面のゲームなのだが，画面構成がうまく調整してあるのでプレイにはなんら問題はない。

白星拾ってごっつぁんです

それでは忘れずに，「出世大相撲」のほうも解説しておこう。これは「エキサイティングアワー」よりもさらに1年くらい前に出た1984年の作品である。それよりも少し前に，「大相撲」というゲームが出ているが，別モノなので，記憶が混乱している人は注意。相撲を取り，勝つことで横綱を目指すゲームである。

ゲームは「エキサイティングアワー」よりもさらにシンプルで，一直線のライン上でジャンプもなく，ひたすら押し合いをするという感じの流れになっている。操作としてはハッケヨイボタンで技を出し，危ないときは貯めてあった「キアイ」を気合いボタンで入れてパワーアップで一発逆転，レバー操作は前進後退に，技の選択となっている。これで，1場所3取組で，勝ち上がっていけばよい。勝ち越せば番付も上がり，負け越せばゲームオーバーである。

このゲームで特徴的なシステムは「根性(辛抱)」メーターというもので，これはまわしを取ったときに登場し，ボタンを連射してこれを満タンにすると技がかかったり，攻撃をしのぐことができるというものである。本来，相撲は呼吸のタイミングで攻めたり耐えたりするのだが，ゲームの感覚としてこのメーターシステムはなかなかうまく相撲の「間」を捉えていて面白いものになっている。

負けたら引退。相撲漫画のような厳しさだ

根性で投げる。連射スティックでもOKか？

相撲と相模は似ている

そもそも「出世大相撲」は，意外に細かいところがよくできていて，取り組み前に表示される対戦相手の得意技を理解して戦わないと，それこそ相撲を取らせてもらえないような一方的な敗戦を喫したり，顔をはたきすぎると，相手が激怒してパワーアップしてしまうといったようなものまである。

ただ，「エキサイティングアワー」のように，この相手にはこうする，というパターンができてしまい，なおかつそれがわかれば一気に勝負がついてしまうので，同じ相手には同じパターンというワンパターンなプレイになり，面白みが極端に薄れてしまうのは，ある意味での限界であると考えるしかないだろう。見た目に単調なパワー勝負である相撲という競技を，ここまでゲーム化しているという点を評価すべきで，これが10年前のゲームであることも忘れてはいけないことなのである。

当時はあまりやっていなかったゲームなので，移植の評価は難しいが，雰囲気は出ているし操作の反応もよく，遊べるものになっている。ただ，これも本来は縦画面のゲームなので画面の感じはオリジナルとは違ったものになっている。「キアイ」マークがいっぱい並ぶと，ちょっと窮屈な印象を受けるが，何かが隠れて困るといったことはないので気分的な問題なのだろう。

過去への執着

さて，このアンソロジーシリーズも第8弾になるが，今回の作品をプレイしてみてシリーズ中最も疑問を感じたことについては書いておこうと思う。確かに知名度ではなく，オリジナルの出来のよさを考えると納得できなくもないが，「これよりも……」と思うユーザーは少なくないだろう。まさかもうネタ切れということはあるまい。

ここ2～3年ほど進歩がなくなってしまったアーケードゲームを見ていると，昔のゲームからは，まだまだ学べるものがいっぱいあるような気はするし，そういった価値のあるものを遊ぶのはよいことである。とはいえ，ゲームはやはり商品であるから，より多くの人間が価値を見出せるようなものを考えるのが本筋ではないかと思うのだが，どうなのだろうか？

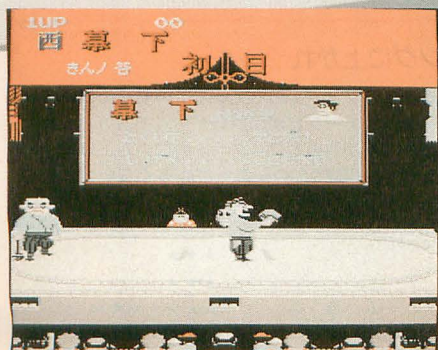
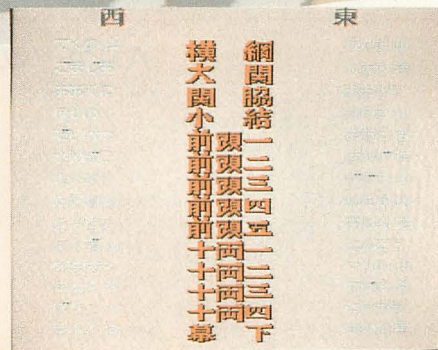
筋肉の祭典

せめて「熱血硬派くにお君」「ダブルドラゴン」あたりにしてほしかったですね。知名度とかが根本的に違うし，事実名作ですから。個人的にはテクノスジャパンは「ザインドスリーナ」に尽きるんだけど，これって「出世大相撲」よりもさらにマイナーかもしれないな。でもいいものはいいのだ，うんうん。

エキサイティングアワー

総合評価	0 ― 5 ― 10
ゲーム性	★★★★★★
技術	★★★★★★
サウンド	★★★★★
グラフィック	★★★★★★
懐かし度	★★★★★★★

出世大相撲

総合評価	0 ― 5 ― 10
ゲーム性	★★★★★★★
技術	★★★★★★
サウンド	★★★★
グラフィック	★★★★
懐かし度	★★★★★★★

相手の得意技を見て作戦を練ろう

番付表。横書きなので威厳がないかも

まっするちゃん，応答せよ

Kiyose Eisuke
清瀬 栄介

女の戦い「レッスルエンジェルス」の続編です。前作の5年後の女子プロレス界。勢力地図は塗り変えられても，目指すは頂点ただひとつ。そして今日もまたひとり，新たな女王の座への厳しい道程を歩み始めるのです。

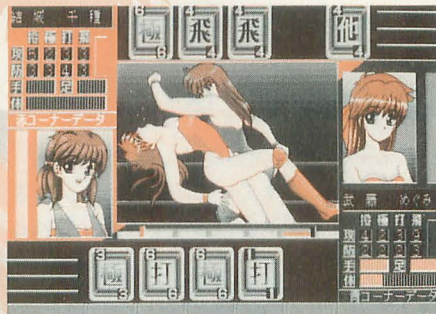
お昼にテレビを見ていたら，プロレスの特集をやっていた。試合じゃなくてプロレスそのものの研究ね。それによると高田延彦のキックの衝撃が重量換算で360kg。普通の人がバットで思いっきり殴りつけても190kgだそうな。ボクのソニックブラストマン最高(比較になるのか?)が90kgだから，プロレスってのはインチキくさそうでいて，やっぱり常人とは別の世界なのだ。

この「レッスルエンジェルス2」は，そういう怖い世界に命をかける少女たちの，汗と流血(?)の物語である。

アタシより強い奴に会いに行く

時は前作の5年後。前作「レッスルエンジェルス」の主人公マイティ裕希子は新日本女子プロレスのチャンピオンに君臨していた。この隆盛を極める新女のリングにデビューしたのが，武藤めぐみと結城千種の2人。だが，おりしも女子プロレス界は再び大きな波に飲み込まれつつあった……。

プレイヤーはめぐみと千種の好きなほうを選ぶことができる。めぐみは飛び技，千種は投げ技が得意なキャラだ。

このゲーム，プロレス画面を見るとカードゲームのようだが，実際にやってみると内容は昔のRPGに近い。というのは，カードの数字よりも，選手の能力値のほうが試合の行方を左右するのである。各レスラーには，投げ技・関節技・打撃技・飛び技の攻撃力と防御力があり，技が成功するかどうかはほぼこの値で決まる。カードの数字はあくまでこれの補正値にすぎない。

髪の色の違いは前作よりもわかりやすい

さらに，相手が弱っていないと大技がかからないとか，相手が立っているかダウンしているかで成功率が違うなど，見た目以上になかなか複雑な計算をしている。それでいて，得意技もカードがないと出せないとか，オールマイティのカードがあるなど，カードゲームのいいところも取り入れられ，実際のゲームバランスはなかなかいい。

ゲームが進んでタッグマッチを組むようになると，戦術も入り組んでくる。このカードは自分が使うか，パートナーに使わせたほうがいいか，どの技なら相手に効率よくダメージを与えられるか，などなど……。

1シリーズ5試合を戦うと，成績によってポイントを得る。これを各能力に割り振り，弱点の補強や長所の強化，あるいは新しい技の習得などにあてるのだ。これはまさにRPGのレベルアップ。負けた原因をちゃんと考えて弱点補強をすると，その結果は次のシリーズで表れてくる。

この成長→戦闘→成長の繰り返しというのが，昔のRPGっぽくていい。能力を上げると試合で試したくなるし，試合をしていると「もっと関節技の防御力をつけたい！」などと次の能力アップが待ちきれなくなる。で，いい成績をコンスタントに残せる頃になるとストーリーが進む。メキシコ遠征への誘い，団体分裂と新団体設立への動きetc.……。メキシコに行くとか，いまの団体に留まるかどうかなどはプレイヤーの選択次第である。

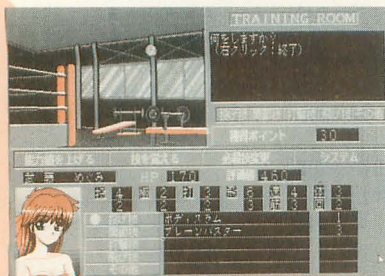
さてさて，この大混乱の行先は……？

プロレスにハマっちゃいそう

女子プロのゲームというから，ミーハーなノリなのかと思ったら，意外に正統派の格闘RPGだった。別に題材が女子プロでなくても面白さはあまり変わらないと思う。

シリーズ最終戦には「水着はぎデスマッチ」があり，倒された女の子が脱ぐ。でも，このゲームの面白さにはあまり関係ない気がするな。そんなに過激な絵でもないし。

演出などに不満はあるが，こういう個性的なゲームは今後も頑張ってほしい。PC-98用は「3」の次が発売間近なので，プログラム的に鍛えてX68000版も続けてほしい。

さて，プロレスの技も覚えたし，今月からWOWWOWにも入ったから，リングスの中継でも見てみようかな。

X68000用　3.5/5"2HD版　4,900円(税込)
プラザー工業(TAKERU)　☎052(824)2493

A級ゲームのリングに上がれ

プログラムと演出は，はっきりいって合格点以下。HDのアクセスは長いし，絵が出るのも遅い。FD枚数もいやに多い。もう少しプロらしさを見せてほしいもんである。それから演出。このゲームはキャラクターへの思い入れが大事なんだから，試合中のグラフィックにはもっと凝ってほしい。相手が誰でもみんな同じような長髪・水着のねーちゃんで，髪と水着の色でしか判別できない。覆面レスラーでさえそうなってるのは許しがたい。同じマシンで12人のキャラがリアルタイムで戦っているご時勢なのに。

ま，おチープだが面白いゲームである。これで作り手が演出とプログラムの能力を上げてきたら，結構RPG界の一角を占めるぐらいにはなれるかもしれないぞ。

総合評価	0　5　10
プログラム	★★★
音楽	★★★
ゲームバランス	★★★★★★★
熱中度	★★★★★★★★

特別編6

ストⅡ世紀末屈辱人生道場

Yokouchi Takeshi 横内 威至

「ストⅡダッシュ」で強いといえば堅いヤツといわれることがある。ハメだとか波動拳固めだとかワンパターンのアレだよ。はっきりいって，試合に勝つが勝負として負けてるプレイだ。見ていてなにも面白くなく，美しくなく，しかも格好悪い。

要するにただひたすらコマンドの正確さを競うだけの対戦なのだ。快感もなく派手さもなく，ただ地道でダークな己との戦いでしかない。俺は決して「ユーウィン」なんていわれるために「ストⅡダッシュ」をやるのではない。だから，俺はこういうハメ野郎とは対戦しない。「ストⅡダッシュ」とは，敵に屈辱を与え，ひとりで快感を貪り友人たちと触れ合う世紀末のコミュニケーション手段である。

やたらと外面のみ，なにごとも格好から入る俺としては，当然見た目美しくないプレイは認めない。ということで，独断と偏見により一部で有名なファンク屈辱攻めが，正しい「ストⅡダッシュ」道であると決定する。まったく使いものにならぬ技をもって相手を倒す。達人だけがなせる芸術であるといっても決していいすぎではない。

ガイルはピエロとなれ

で，今回は俺が「ストⅡダッシュ」で好みのファンク野郎，ガイルを解説しよう。一見，地味野郎の代表格であるが，どの角度から見てもカッチョ悪い技の数々，オリジナリティあふれるヘアスタイル，類希なるボディコンメタル軍服，ファンク道としては避けようのない魅力でシビレさせてくれるのだ。ちなみに眉毛もない。

まず，ガイルの基本といえばソニックブーム。ソニックブームを追い掛けていってしゃがみ中キック。地味すぎてカッチョ悪くてもう幻滅。できるかぎり避けたいプレイである。では追い掛けていって跳び込み→「なんとか」。「なんとか」には大キック，中キックなんかが定番。でもやはり地味だからあまりやりたくない。例外として，中キックが「ガイルキック」という恥ずかしい名前であることを互いに理解していればやや屈辱的。恥ずかしくないか？　たとえば「横内パンチ」「川田チョーパン」なんてのを食らったら，あんた恥ずかしくて痺れるぜ。まあ妥当なところは小キックであろう。なるべくソニックブームは使いたくないが慣れるまで辛抱するしかない。攻め始めがソニックブームしかないのがガイルの欠点であるが，がんばって派手に美しく，あるいはカッチョ悪く屈辱的にいこう。

では跳び込みが決まったらどうする？　まさか小パンチ（キャンセル）→サマーソルトキック？　シケてるね。せめてカッコよく，アッパー→サマーソルトキックに切り替えたほうがいいぜ。なお，これはガイルで数少ない正統コンビネーションとして存在するもので，このほかにも4段攻撃がある。4段は，跳び込み→立ちアッパー(キャンセル）→ソニックブーム→裏拳。しかし，アッパー→サマーソルトキックは，サマーソルトキックコマンド途中のレバーがニュートラルの瞬間に大パンチを喰らわしておく超難易度の技だ。堅いガイルでもこれぐらいはマスターすべきである。

喰らえ！　必殺ガイルキーック（恥）

では，よりファンクな技をマスターしようではないか。まず，ファンク技として好んで使うのは中パンチ。間合いが遠ければファンクアッパー，近ければただこねパンチだ。あとはニーバズーカー。そして対空兵器としてサマー失敗垂直ジャンプ中キック。次にコンビネーションとしての使い方だ。まず4段の変形で，最後の裏拳の代わりにファンクアッパー。これでピヨらせればもう無敵。続いては3段（アッパーサマー）の変形でアッパーの代わりにやはり中パンチ絡み。ファンクアッパーサマーがかなりシビれる。これらのファンク技は絶妙に対空兵器としても使えるため，マスターすればかなりのファンクになれる。

かつてはガイルの冠詞として「待ち」「クソ」「ハメ」などが存在していたが，たまにその要素を加えてやるとかなり刺激的。勝負が熱くなるぞ。たとえば対空兵器として立ち中キックだとか，禁断のソニック投げなんかが素晴らしく泣ける。もうひとつはズンズンガイルなんかもシブい。これは端に追い詰めて小キック，小パン，キャンセルソニック，跳び込み小キックなんかで攻め固めるガイル。たまにハメるとけっこう屈辱的で燃え上がること間違いなし。

ガイル，男の花道

決してソニックを多用することのないように。投げは空中投げ以外絶対禁止。もちろんほかのキャラでも同様のマナーだ。飛び道具イコール下衆野郎，投げ技イコール人間失格と思われるのが我々のストリートでの常識だ。より多くの人間が屈辱プレイに目覚める日を俺は待っている。

跳び込んできた相手にはサマーソルトキックだ！

跳び込み大パンチからの連続技をマスターせよ

特別編7

バイソンの魅力を探る

Nishikawa Zenji 西川 善司

M.BISON

連日★型の剃り込みを入れに床屋へ殺到する群集，そして街に溢れるボクシング・ファッション。バイソン人気はとどまるところを知らず社会現象にまで発展しそうな気配である。

バイソンはジャンプ力に乏しく，飛び道具型の技も一切持たず，基本的な戦闘能力は，ほかのキャラクターと比較すると見劣りしているといわざるをえない。しかし，最強の必殺技ともいえるターンパンチの破壊力は凄まじく，これが生み出す一発逆転の爽快感はバイソンでなければ絶対に味わえない。このあたりが街のゲームフリークたちの心をつかんだのだろうか。

バイソン2通りの戦法

キックボタンを3つ，もしくはパンチボタンを3つ同時に押し溜め，3つのボタンを放した瞬間，溜めた時間に比例した破壊力を持つパンチが繰り出される。これがバイソンの必殺技ターンパンチである。バイソンを使って闘いを挑む場合は，このターンパンチが要となる。

バイソンの戦い方には2種類の方法があるとの報告が，日本バイソン学研究所から寄せられている。ひとつは絶えずターンパンチを溜めておき，通常の戦闘は残された3つのボタンのみで戦い，ここぞというときにそれまで溜めていた強力なターンパンチを打ち込むというもの。もうひとつはほかのキャラクターと同様の，6つのボタンを駆使した戦いのなかでチャンスがあればターンパンチを繰り出すというものらしい。

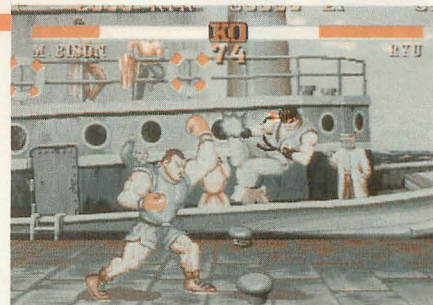
このレポートをもとに，まずは前者の方法から紹介していくとしよう。

玄関開けたら2分でターンパンチ

まず，そのボタンと指の対応だが，親指で小キックボタン，人差し指で中キックボタン，小指で大キックボタンを押さえる。この3つを絶えず押すことによりターンパンチを戦闘中いつでも出せるようにしておくのである。そして中指と薬指をそれぞれ中パンチ，大パンチに割り当て，通常はこれら2つのボタンで戦うことになる(小パンチは指が届かないので使えない)。

キックボタンを3つ押しっぱなしにしている都合上，ダッシュアッパーは使えないが，ダッシュストレートは大，中のバリエーションで繰り出すことができる。中パンチによる跳び込み攻撃と逃げ攻撃はできるし，つかみ技の頭突き攻撃も可能。さらに，飛び道具で端に追い詰められたときに突破口を開くことができる大パンチ・ストレートも健在である。

対空にはしゃがみ大パンチ(アッパーカット)がある。これはほとんどの跳び込みを撃墜できる非常に強力なものだが，技が出るまで多少の遅れがあるため，敵の跳び込みを感じたら早めに反応する必要がある。

2つのボタンだけで戦闘を行うというのは一見無謀に思えるが，このように中パンチと大パンチだけで基本的な攻撃はすべて可能であるため，ほかの4つのボタンが使えないことはそれほどハンデにはなっていない。バイソン学の権威である梅村教授は逆にこの戦法を「むしろ溜まりに溜まったターンパンチをいつ食らわされるかわからないという心理的抑圧を絶えず相手に与えることができる」と分析評価している。

竜巻旋風脚だっておとせちゃうんだよ

6ボタン戦法の利点

6ボタンを用いた戦法は，小攻撃の連射が可能な点，そして相手の動きに対応した適切な対応が随時可能な点が，ターンパンチ戦法にはない特長といえる。また，こちらの方法だと，ダッシュストレート→ダッシュアッパーをからめた連続攻撃も行うことができる。対戦プレイにおいては，相手のキャラクターの性能や相手プレイヤーの戦法に応じて，2つの戦法を随時選択しなければならない。

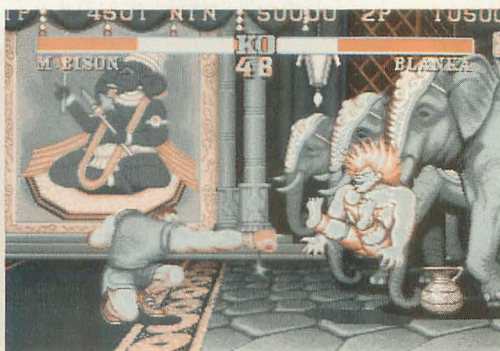
あと，しゃがみ大キックで出せるしゃがみストレートのパンチの先端はほぼ無敵で，写真のようにブランカの電撃などもはじき飛ばすことができる。これは6ボタン戦法を使う場合にはぜひとも頭に入れておきたいポイントだ。必ず相手を転ばせることができる点も特筆に値する。

腕の先っぽはもうほとんど無敵クレイジー

当たれば必ず相手を転ばすことができる

バイソンフリークへ捧ぐ

バイソンは，ストII登場キャラのなかでも最も気絶しにくいキャラクターといわれる。このあたりを念頭に入れ，ターンパンチの一発逆転性を活かした，最後まで勝負を捨てないブラッディな闘いを展開してほしい。

バイソン評論家　西川善司

特別編8

とにかく勝ちたい人のために

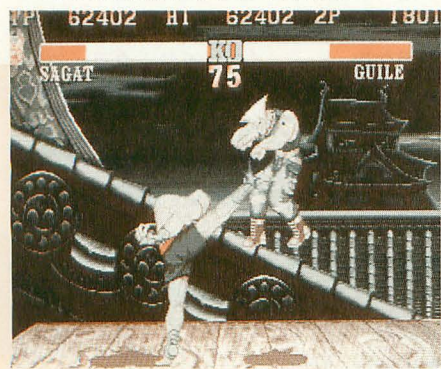
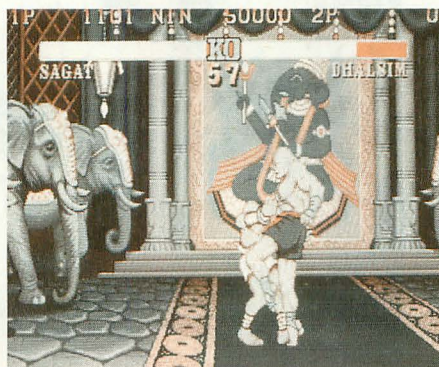
Taki Yasushi 瀧 康史

本当のことをいうと，私自身はあんまりこのサガットは好きじゃない。なぜって……強いから。なーんにも考えなくても強力なキャラで面白くないんだもの。初心者向けのリュウ，ケンだといえるかな。

だから，とにかく勝ちたい人や，一緒に遊ぶ人が強すぎて自分は負けてばかりいるなんて人は，このサガットを使うとよいでしょう。逆に自分がうますぎて対戦仲間がかわいそうなときに，相手にサガットを使わせて，自分はザンギやバイソンやケンなんかで遊ぶのもよいかもしれません。

跳ばせて落とすのがセオリー

必殺技の性格からわかるように，サガットは跳ばせて落とすリュウ，ケンタイプ。6種類のタイガー(主に下)を巧みに使い，読まれないように敵に跳ばせ，まんまとタイガーアッパーカット(以下アパカ)で落とすというのが主な戦略パターン。ただし，本田やベガなどには，下タイガーが突進技ですりぬけられてしまうので，あくまでも次のタイガーが，上か下か，はたまた速いか遅いかを読まれないようにすること。

また，昇龍拳とは違い，アパカの無敵時間は出始めしかないので，できる限り引きつけて撃たないと，最悪の場合，一方的に食らうことがあります。コマンド入力をして斜め下で止めておき，しゃがんでいる状態でタイミングよくボタンを押すと，わりあい引きつけが成功しやすくなります(ケンの1000点昇龍拳もそれと同様)。

強キックは早めに出すこと

小キックをキャンセルして……

ただし，完全にアパカが成功しても，めくられてしまうと，昇龍拳と違って反対側を向いてくれないことがよくあります。そうなると，あとは敵の思う壺なので，注意が必要です。また，意外にもダルシムのしゃがみ中パンチやリュウ，ケンの遠目の大足払いに一方的に負けてしまいます。

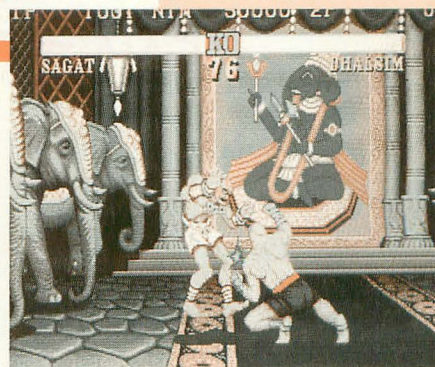
サガットにはアパカ以外にも強力な対空技がいくつかあります。遠いときは大足払いで着地点で転ばせ，ほかにも立ち大キック，立ち大パンチなどがあります。立ち大キックは出るのがわりと遅いので，早めに出す必要があります。もちろん，間合いとタイミングが悪いと相打ちになりますが，もともと強いキャラだし，対空はたくさんあるので，アパカに固執せずにいろいろ使ってみるほうが美しいプレイといえるでしょう。

なお，半分卑怯な技かもしれませんが，起き上がりにアパカを出すと，相手が防御していても攻撃を当てることができます。

イニシアチブをどう取り返すか？

対戦ゲームにおいて自分がイニシアチブを取ることは，勝つためのポイントです。

これはすべてのキャラでいえることですが，いかにして取り返すかはキャラごとに違います。多くの場合，キャラの長所を活かした攻撃をし，弱点をさらさないというのが，その一般的な方法といえます。

サガットは食らい判定が体に比例して大きな分，逆に背が高いのでめくられにくいという利点があります。また，これも背が

タイガーアッパーカットを出す

高いことに影響しますが，食らい投げ*1がやりやすいことを忘れてはいけません。

長所といえば，跳び込み大キックや，大パンチの射程が，インチキなほど長いので，リュウ，ケン相手には，波動拳を確認したあとでも間に合ってこちらの蹴りが入ります。

タイガークラッシュで，相手に近づいたときにも，落ち際が思っているほど弱くはないので，こちら側がアパカや投げを入れるとなぜか決まります。相手は近づかれたときに，足払いや投げなどをしようとして逆に当たってしまうからです。

実は投げの間合いが短かいのと，足が遅いのと，ハゲぐらいしか欠点がないのです。

連続技

必殺技でキャンセルできる通常技は，立ち小キックとパンチ，しゃがみ小キックとパンチです。立ち小キックは2段技なので，1度目をキャンセルします。

したがって，連続技の数にも限りが出てきます。メジャーなのは，跳び込み大キック(or大パンチ)，立ち小キックキャンセル，アパカ。ほかにも跳び込み大キック，立ち中キック，しゃがみ小キック，大クラッシュ(orアパカ)などがあります。これなんかは立ち中キックとしゃがみ小キックのつなぎが結構難しかったりします。

もっとも，これらに挑戦するような時期になったら，そろそろサガットも卒業すべきころかもしれませんね。

*1 相手から跳び込み技を食らったとき，その打点が高ければ，相手が落ちてきたときに投げることができるというもの。これによって相手を転ばすことができるため，イニシアチブを取り返すことができる。

1993年度 **GAME OF THE YEAR** [自由応募部門]

1993年

愛されたキャラクターたち

1993年のゲームには，個性的で魅力的なキャラクターがたくさん登場しました。

なかでも女の子が人気独占というのは予想に違わずといったところですが，意外だったのは格闘系や脱衣系は案外人気薄だったこと。美人タイプの彼女たちよりも，人気を集めたのは，コットン，シルク，リルルの「可愛くって，ヘンな性格」の3人娘。明るく元気で，脳天気。しかもちゃっかりしてる，ってのが共通点。も，もしかしてX68000ユーザーってみんな，尻に敷かれるタイプなんじゃ……？　男ではなぜかチェルノブがダントツの人気でした。

では，特に人気の高かったキャラクター5人に寄せられた声を紹介しましょう。

▲上村　珠緒　愛知県

▲近藤　陸生　埼玉県

▲青木　一師　奈良県

シルク　（コットン）

▶「コットン」がゲーセンにあった頃，私の大きな目的はティータイムのあとに出てくるBigでGreatなシルクちゃんでした。女性に多大なあこがれをもっていた(女に飢えてたともいうか？)当時，彼女のツヤツヤのおみ足を見たいがために100円をつぎこんだことも……(なさけない)。あの思い出を込めて，この1票を贈ります。

桑原　秀　兵庫県

▶適当なことをいってコットンをだますいじ悪さ。いい味出してますねぇ。

森崎　剛(21)広島県

コットン　（コットン）

▶あの戦いの理由が食欲のためだけというのが素晴らしい。　須藤　祐一(27)香川県

▶彼女がいなければ，このゲームも十把ひとからげのただの横シューティングと化したでしょう。　石澤　清光(30)東京都

▲鯉江　将弘　愛知県

▲清水　健太郎　東京都

リルル　（エトワールプリンセス）

▶服装のせいか首がないように見えてしまうが，かわいいから許してしまおう。

田沼　基司(27)茨城県

▶なんたって，リルルの性格がいいっ。こんなヤツがオレは好きだ。

吉田　晴彦(18)新潟県

人形　（悪魔城ドラキュラ）

▶「おまえは誰？」「帰りなさい」「やめてっ！」「いやあっ！」……隣の部屋の弟の動きが一瞬止まる。エロゲームやってる，などと思われてたりして……。ああ，兄貴の尊厳がぁ！　(そんなもん，もとからなかったりして)　山西　孝到(19)大阪府

▶「おかえりなさい」
「ただいま」オイ。

伊藤　千光(20)千葉県

▶4匹まとめて近づいてきたときは背筋が寒くなった。　浪越　孝宏(21)兵庫県

▶アパートの隣人の誤解を招き，それ以降の隣人関係がまずくなった。

大畑　佳史(20)兵庫県

チェルノブ　（チェルノブ）

▶「ペルシャの国の王子様」より美しい体の動き，あと戻りできない人生，悲しいエンディング。ああ，君のことは(忘れたくても)忘れないっ！　古沢　達也(21)埼玉県

▶Cボタンを連射するだけで笑えるのに，さらに空中まで歩いてくれる彼はとってもおちゃめ。　河野　裕文(18)静岡県

▶戦う人間発電所にこの賞をあげなくて誰にあげるのだ。　越智　亮(21)大阪府

▶世界の全ゲームにおいて，唯一のヒーローキャラ。悪夢のような現実に悲痛な決意で立ち向かう彼。そのあまりに哀しい結末ゆえに彼が忘れ去られることはない。

高橋　明(23)東京都

そのほかのキャラクターたち

メデューサ　（悪魔城ドラキュラ）

▶メデューサちゃんは「私にとっての女性の理想像」です(笑)。やっぱり彼女を倒すときは胸が痛みます。上手な人は，彼女の体力を残り1ポイントにして彼女が画面中央近くでジャンプしたときに同時にジャンプして倒しましょう。音楽が止まり，互いの体が空中で交差して，同時に床に落ちる。そして，彼女がくずれてゆく……。映画の1シーンを見ているようで感動します(かわいそーでしかたなかったりする)。

平野　鉄之助(18)長野県

階段　（悪魔城ドラキュラ）

▶何度あのザンギのスクリューより広い吸い込みに泣かされたことか……。

橋本　健一郎(24)神奈川県

シラケ鳥　（クレイジークライマー）

▶哀愁漂うBGMと共に現れ，フンをまき散らして去っていく……。なぜかこいつにやられても腹がたたないんですよね。

西嶌　郁夫(25)大阪府

審判　（ぶたさん）

▶地に沈み，イエローカードを出す。もう，彼しかいない。曲木　一博(19)栃木県

湯飲み　（コットン）

▶湯飲みが降ってくる夢をみた。

杉山　浩一(23)長崎県

読者レポート

その1 ゲーム体験談「エトワールプリンセス」

X68000をもっていながら，極力シューティングには手を出さないようにしています。これは単にRPGが好きだからという理由だけではありません。

要するに私はとろいのです。たとえば，「イースⅢ」で薬草，精霊の首飾り，指輪の回復力も半分以上は残っています。にもかかわらず，28戦28敗，いまだにエンディングは見られません（笑ってやってください）。

だからアクションRPGと聞いて購入に二の足を踏んだのですが，「初心者でも大丈夫」と某雑誌で紹介されていたので買ったのがこれ「エトワールプリンセス」です。

攻撃が魔法主体というのも嬉しいかぎりですし（武器による接近戦は苦手なもので），グラフィックの美しさにはただため息。教会のステンドグラスやお城，草木に至るまでその色合いは素晴らしいものです。多彩な魔法のなかでとりわけ気に入ったのは，火と土の女の子が使う通常攻撃（買おうとしたときは，すでに発売中止になっていたので泣く泣くTAKERUで購入。詳しいマニュアルがないので女の子の名前がわからないのです。悲しい……）。あのぽわぽわとした火球と，3方向に流れる光，実にきれいでしょっちゅう見とれてました。

ほかにも水の中を歩くときは浮力を感じるし，鏡を回すときの効果音などもいかにもそれらしいしで，どれひとつとっても丁寧に作られたことがうかがえます。そして決定的なのは，この絵本のような世界にかっとんだギャグ！　キャラクタは可愛いけど，どこかとぼけたところがあって，何度も笑いを誘われました。敵役である魔女といえども同様で，あのセンスにはもう爆笑ですね。まねき猫やら「根性」の看板やら……それに「仮面ラ○ダー」の変身ポーズ（歳がばれるなあ）。

でも，私にとって特別ありがたかったのは，このゲームシステム。落ちても怪我ですむ，ボスに負ければ直前からやり直しがきく，時間や回数に制限がないなど，あちこちに工夫が施されているので，物語を楽しむ余裕があったのです。それではあっさり解けてしまったかというと，これが全然（笑）。

ボスキャラのうち，一度で倒せたのはドラゴンのみ。あとは最低3回から最高10回はこっちが倒れましたっけ。ドラゴンとて偶然に手の甲に乗ったからでして，それに気づかなければ，やり直しの連続だったでしょう。ほかにも道に迷うわ（地図があってもこれだ），罠に引っ掛かるわ，連続ジャンプにしくじるわで，何度もゲームオーバーになったものですから，エンディングを見られたときは感動しました。よってこれを推薦します。　黒田由紀子(29)群馬県

その2 Dear KONAMI

「悪魔城ドラキュラ」。初めて出会ったのはもう8年前になる。当時ガムシャラにファミコンのゲームを征服していた自分だったが，なぜかドラキュラだけはただならぬ印象を受けた。それゆえに学校では，壇上から「せむし男」ごっこをして飛び降りて足を捻挫し，「シモン・ベルモンド」ごっこでビニール製の縄跳びを振り回して腕にミミズばれをたくさん作っていた。

幾年かのブランクが過ぎ，ドラキュラに再会した。この間に自分のゲームに対する見方，感じ方などは大きく変わってしまっていた。しかし，再び憎きドラキュラ伯爵を倒したとき，あの当時の捻挫した足の痛み，腕にはったミミズばれのおかげで風呂に入るたびにヒリヒリしていた感覚が懐かしく思えてきたのだ。

近年，ゲームの表現技術や演出には目を見張るものがある。もちろんX68000版も高度な技術力と演出でプレイヤーを驚かせたが，もうひとつ，深い思い出がある自分には卒業アルバムのように過去の姿を鮮明に浮かび上がらせるのも素晴らしい演出効果のひとつだと考えた。

企画，制作元のコナミではこのような相乗効果を狙っているわけではないだろうが，これも熱心に長年ゲーム開発を続けている結果の証であり，賜である。こういう経験を味わったのは自分ひとりではないはずだ。こんなユーザーがいるかぎり，コナミさん，これからもがんばってください。

鈴木　政宏(20)宮城県

MAKER OF THE YEAR

素晴らしいゲームで私たちを楽しませてくれたメーカーさんへの感謝の声も，いろいろ寄せられました。なかでも多かったのは，この2社です。そこで，今後への期待も込めて，1993年もMAKER OF THE YEARを贈りたいと思います。

魔法株式会社

（餓狼伝説，餓狼伝説2）

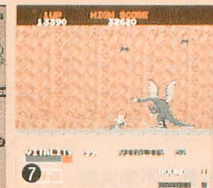
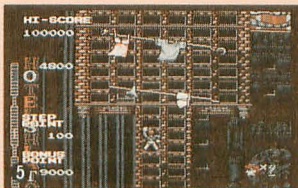
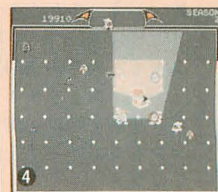
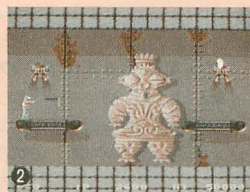
▶1993年の魔法株式会社のがんばりは本当にうれしい。そして今年の春，夏に「龍虎の拳」「餓狼伝説SPECIAL」と立て続けに発売が予定されている（「侍スピリッツ」も出してほしい）。そこで，ソフトハウス・オブ・ザ・イヤー大賞は，魔法株式会社だ！

石井　一士(22)千葉県

電波新聞社

（ビデオゲームアンソロジーシリーズ）

▶短期間に，あれだけのクオリティの高いゲームを，しかも安く提供してくれる電波さんは偉いっ！　これからもこのペースでがんばって～。　秋田　年弘(21)奈良県

1	テラクレスタ/ムーンクレスタ
2	チェルノブ
3	スターフォース
4	リブルラブル
5	クレイジークライマー/クレイジークライマー2
6	ぶたさん
7	ドラゴンバスター
8	エキサイティングアワー/出世大相撲
9	アルゴスの戦士

勝手に GAME OF THE YEAR

みんなの意見とは違うかもしれないけれど，ちょっと気になるこのゲーム。誰がなんていったって，心に残るはそのゲーム。大きな賞は似合わないけれど，お気に入りのあのゲーム。さあ，自分だけの特別賞をあげちゃうのだ。

リサイクルな人々

インストールしま賞　餓狼伝説2
▶サンプリング音を吸い出して，自分の環境に組み込むと……。
浜田　淳(20)大分県

効果音賞　コットン
▶パソコン再起動時に「ふっか～つ」といわせたりして有効活用しているのは私だけではない……と思う。
植木　正幸(25)神奈川県

仲良しな人々

ドラキュラよりも強いで賞　悪魔城ドラキュラの狼女
▶どうしても倒せないので，弟に助けてもらった。　永尾　健次(25)大阪府

主演キャラクター賞はスーパーリアル麻雀PⅡ＆PⅢのショーコの靴下で賞
▶うちの妻(26)をして「もったいつけてんじゃねーよ，このアマ」と叫ばせた。いいからコントローラ返せ。

みんな仲間で賞　ストリートファイターⅡダッシュ
▶発売当日，九十九電機のエレベータの中で，全然知らない人と「ストⅡですか」「もちろん！」と会話できた。
腰原　仁志(31)神奈川県

???な人々

ものすごい圧縮技術で賞　餓狼伝説2
▶100メガショックのゲームが6メガに……。NEO・GEOめ，本当に100メガも使ってんのか。JAROに電話するで！
矢田　岳雄(19)石川県

勝手にプログラミング技術賞　ロボットコンストラクションR.C.の松下哲也氏制作のHDL-3 2足歩行型ロボット
▶……え，ちがうの？
松下　哲也(20)兵庫県

ストⅡ左向きの法則で賞　ストリートファイターⅡダッシュ
▶中級者の私は左側を向いているとき，コマンド式のキャラを使うと急に弱くなってしまう。　石崎　将希(19)茨城県

満足でしあわせの人々

行動が自由で賞　エトワールプリンセス
▶川へも入れる，カベを越えた攻撃もできる。そうだ，いままでのRPGが変だったんだ。
渡辺　久孝(26)大阪府

かゆいところに手が届くで賞　コットン
▶コンフィグのセーブ機能が最高！　他社も真似してほしい。
石井　英一郎(20)千葉県

オーッホッホホホホホホホで賞　ああっ！お姫さま！(The World of X68000)
▶パソコンのゲームって難しくって，敵が堅くて，どれもこれもボロボロで，投げ出したくなったとき，このゲームに会えてよかったぁ。いままでのこと(ドラキュラなど)が嘘のように，晴ればれ日本晴れだぁ。
鈴木　俊雄(25)福島県

頬擦りスリスリしたいで賞　ぶたさん
▶白ぶたくん好きぃ～！
山谷　尚(19)大阪府

作況指数100%で賞
▶春麗，不知火舞，コットン，エトワールプリンセスの女の子たち……etc, etc。……いや，単に今年は女の子が「豊作」だったな～と。　池田　譲太(25)大阪府

現在の女子プロレスはこんなもんじゃないで賞　レッスルエンジェルス
▶でも，その時代錯誤な感覚が心地よいのです。　春名　義行(27)兵庫県

ジョイパッドがすずなりで賞　チェルノブ，リブルラブル，ストリートファイターⅡダッシュ，餓狼伝説2
▶この4本のおかげで，うちのX68000のまわりはジョイパッドでいっぱい。
河野　太郎(20)東京都

サルな人々

基本は血へど吐くまでで賞　スターフォース
▶腱鞘炎になったとき，世界の向こう側が見える。　松本　款一郎(23)石川県

あれは夢だったので賞？　スターフォース
▶半分眠りながら100万点ボーナスを出したような……出さなかったような……。気がついたら，おでこにキーボードのあとがついてました。　岩瀬　貴代美(22)福岡市

苦労の人々

相手にちっとも巡り合えないで賞　ロボットコンストラクションR.C.
▶自分のロボット同士を戦わせると特に……(泣笑)　松本　拓司(19)埼玉県

許せないで賞　鳳凰脚
▶ガードしてもあれだけゲージが削られると……ナットクいかーん！
中矢　史朗(23)愛媛県

どうなってるんで賞　ヴェルスナーグ戦乱
▶いちばん初めの盗賊の洞窟がクリアできない。レベルは12，金は1755……。誰か教えて！　(半年間もこのまま)
江城　憲之(18)大阪府

そのほかな人々

おすすめゲームで賞　ダイナミックフォーメーションサッカー(TAKERU同人ソフト)
▶ワールドカップ予選以降フラストレーションが溜まっていた私の前に突如現れたこのゲーム。いやーハマるハマる。音響関係がショボイのを除けばかなりの出来栄えだぞ。値段も2,000円と安いことだし，さあみんな，最寄りのタケルへ直行だ！
松本　拓司(19)埼玉県

ホントは大賞で賞　○級生
▶そうだよねっ，みんな！(笑)
堀井　晶司(21)北海道

えむな人々

もっとイジメてほしいで賞　スーパーリアル麻雀PⅡ＆PⅢ
▶ゲーセンでの借りはいつになったら返せるようになるんだ。
久保田　忠弘(32)埼玉県

最後に今週の困ったちゃん

大失敗で賞　2月号GAME OF THE YEARノミネート校正者
▶何が1で，何が2だかわからない。
秋山　欣之(28)広島県

ああああ，ごめんなさあい。結局，ほとんどの方が賞名を併記したり，「去年と同じ」との注意書きつきで投票してくださいました。川よりも長く，海よりも深く反省してます。　(担当者)

ほんの一な人々

1回クリアしてから一度もやってないで賞　スーパーリアル麻雀PⅡ＆PⅢ
▶すみません。邪念だけでやってました！
金渕　満(17)青森県

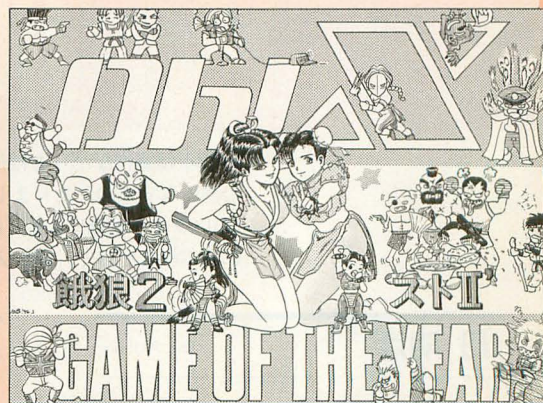
ぼよよん賞　不知火舞
渡辺　貴公(23)福岡県

ぷるるん賞　不知火舞
辻野　雅弘(21)滋賀県
▶舞ちゃんのボヨヨンはNEO・GEO比150%(当社比)。　敏森　健裕(21)兵庫県

ILLUSTRATION GALLERY

▲岩瀬　貴代美　福岡県

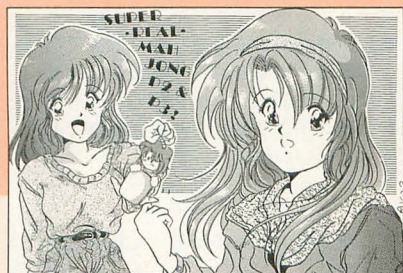
▲芹澤　あや子　東京都

▲鈴木　貴久　神奈川県

▲折坂　信春　大阪府

◀高橋　明　東京都

▲冨士井　淳　岐阜県

▲大高　孝平　宮城県

▲藤澤　篤　奈良県

▲横井　賢一　富山県

▲姉帯　寛　茨城県

▲尾形　雅治　京都府

▲武田　和凡　京都府

▲佐藤　貴是　神奈川県

▲新川　洋司　京都府

▲小川　伸輔　宮城県

▲上田　考一　福岡県

▲占部　哲彦　広島県

Oh!X スタッフによる

勝手に GAME OF THE YEAR

1993年は寡作ながらも粒ぞろいのゲームが発売され，Oh!Xに寄せられたメッセージにも，心熱くなるものがありました。そんな1993年度GAME OF THE YEARのしめくくりは，Oh!Xライター陣による「勝手にGAME OF THE YEAR」です。たくさんのゲームを実際に遊ぶ機会が多く，思い入れ度も高い人が多いスタッフたちの声を聞いてみましょう。

丹 明彦 Tan Akihiko

1．悪魔城ドラキュラ
2．ストリートファイターⅡダッシュ
3．なし

ちと無難か。

「悪魔城ドラキュラ」は久々に達成感というものを教えてくれたアクションゲーム。絵，音，演出はプロフェッショナルの力量を感じさせ文句なしの出来だが，私としては，やり遂げるまでの苦難の日々に対して1票を投じたい，たとえ私には1周が限界としても。

「ストリートファイターⅡダッシュ」はやや遅かったかなという気がしないでもない。移植作品としては素晴らしい。初代機でも立派に動くといえば聞こえはいいが，初代機から根本的に変わっていないX68000を思ってややもの悲しくもさせられる。ほかにはほとんど遊ばなかったので3位は割愛。

アーケードでひとつ推せといわれれば「バーチャファイター」。「リッジレーサー」も捨てがたいが，私が100年かかってもたどり着けそうにないという観点で選んだ。データの精巧さもさることながら，セガの見せ方のうまさには感心させられる。硬直化した格闘ゲーム界にもたらされた強烈な一撃という点でもポイントが高い。いずれにせよ，私はアーケードでは3Dものしか評価できないらしい。

昨年は海外作品にこれはというのがなかった気がする。AMIGAなんていうCD-ROMに乗り遅れた機種を使っているのも原因だろう。が，CD-ROMが普及した結果，ゲームデザインの方向性が私の好みでないもの，つまり無数のイベントを記憶容量でカバーするという方針に寄ってしまったせいでもあるのだ。私はプログラマとして感動したいのだよ。

進藤 慶到 Shindo Noriyuki

1．悪魔城ドラキュラ
2．ドラゴンバスター
3．DIVE ON

うーん，「ドラキュラ」以外に1位をつけることのできるものが見当たらない。理由は過去のレビューを読んでもらえば明らかなとおり。コナミの魂を感じる作品だった。すべての面において原作に勝っていなくてはグレードアップとは呼べないと，これ見よがしに，かつ，さり気なく見せつけている(なんつう表現や)恐ろしい作品だ。

「ドラゴンバスター」は，かつて狂うほどプレイしていたということもあり，どうしても外せない逸品。いつもながらの確かな移植により，安心してあの頃の思い出に浸れるのは嬉しい。音源の違いを超えた，曲の完成度にも拍手。なんといっても曲が重要だからね。

3位の「DIVE ON」は1993年10月号のTHE USER'S WORKSで紹介されたシューティングゲームだ。難易度が急上昇気味のアーケードシューティングにいささか憤りを覚えていたところ，これである。どこかで見たような気もしなくはないが，ツボはそこそこ押さえてあり，なにか光るものを感じた。個人的には打ち込み感覚と熱い稼ぎをさらに追求したシューティングを望みたいところ。とにかく私はいまシューティングに飢えているので，「2」への期待票の意味も込めて3位。

まあほかにも，信じられないほど忠実移植の「ストⅡダッシュ」や，トレースプレイを究めるのが楽しい「コットン」など捨て難いゲームもたくさんある。数こそ少なかったかもしれないが，昨年度は良質の作品に恵まれたいい1年だったといえるかもしれない。気がかりなのは，「これは！」というオリジナルが少なかったことか。そういった意味では「ジオグラフシール」はたいへん楽しみだ。まだX68000はイケるぞ。

私事では「アウトランナーズ」が熱かった。ゲームであるがゆえに許される走りと豪快さ，気持ちのよさはシリーズ健在である。この手のゲームを作らせるとやっぱりセガだなぁ。X68000シリーズにも，せめてこんなのがスイスイ動くような新機種でも出てくれればいいのにな。

瀧 康史 Taki Yasushi

1．コットン
2．ネメシス'90改
3．ぶたさん

実は，最近は，あんまりX68000でゲームしてないんだよね。こんなこというと反論がバシバシきそうだけど，そろそろX68000でゲームをする時期は過ぎてるのかもしれない。大半のゲームはゲーム機でやったほうが便利だし，出来がよい場合が多いから。

それでも，X68000のゲームには出来がよいものが多い。基本的なお勧めは上に挙げた3つ。シューターだからな，私は。「コットン」は「You!Do!」モードがお勧め。「ネメシス'90改」も玉避けが熱すぎてGood！ 個人的には，「悪魔城ドラキュラ」とかにも，票を入れたいところだけど，実はやってないのだ(はたでは見てたからよいゲームというのはわかってるよん)。「ぶたさん」は，多人数でやるとおもしろいんだよね。4人プレイとか5人プレイとかあったら燃え燃えなのにね。

話は変わって，ここのとこ，X68000にゲームを作ってくれるメーカーが減ってきてるけど，それは当然かもしれないね。PC-98があるのにわざわざX68000で発売する必要はないし，コンシューマ機で遊べるゲームを移植してきても，たいした売上げは望めないし。だってX68000ユーザーって，割とほかのゲーム機やパソコンもってる人多いからさ。つまるところ，ゲームメーカーはX68000での売上げを望むなら，X68000オリジナルぐらいしか望めないね。そういう面で，残ったものは条件を兼ね備えているともいえる。「ドラキュラ」しかり，「コットン」しかり，「ネメシス」しかり……。

個人的には，「ダンジョンマスター2」が欲しいところだけど，やっぱり初めに出たPC-98でやっちゃうもんなぁ。SXで出るなら，話は別だけど。

今年はオリジナリティ溢れるゲームが出るといいな。

清瀬 栄介 Kiyose Eisuke

1．ストリートファイターⅡダッシュ
2．ロボットコンストラクションR.C.
3．項劉記

ゲーム界ってのは，流行の本筋をいくゲームがドンとあって，その周りにいくつか亜流があって，さらにその周りには我が道をいくソフトハウスが，次の流行を奪い取るべく試行錯誤しているという構図が理想だと思う。その意味では，今年のX68000はなかなかいいセンいってたんじゃないだろうか。

今年の流行の本筋は格闘ゲームだったからして(ここ3年ぐらいそうだけど)，やっぱり「ストリートファイターⅡダッシュ」をいちばんに推さないわけにはいかない。こいつはひとつのスタンダードである。せっかくX68000をもってるんなら，ウチに置いてしゃぶりつくしたい。ただ，ボクはもう「ストⅡ」式のゲームルール(コマンド式の必殺技，ジャンプで跳び込み，レバーを引いて防御など)には飽きちゃった。ゲームセンターでも「スーパーストⅡ」より「バーチャファイター」のほうが新鮮で面白いと感じてしまうので，このテのソフトが続々出てきても支持はしない(おぉ，高飛車)。「マッドストーカーX68」ぐらい毛色が変わっているものだったら，新鮮で遊ぶ意味があるんだけどね。

いつものことだが，アクション以外のジャンルはコマ不足気味である。もうこれはPC-98の性能の上でしかRPGやシミュレーションが企画されていないせいというしかない。そんななかで「ロボットコンストラクションR.C.」の登場は歓迎したい。ゲームの骨格は大昔ログインでやっていた「ロボットバトルなんとか」を思い出すが，X68000上で作り直され，AV機能を活かしたインテリジェントゲームという珍しいスタンスの作品になっている。

「項劉記」もアクション以外のジャンルの作品のなかではチャレンジングなゲームデザインをしているところが好感をもてた。「三国志」シリーズみたいな「イチゲンさんお断り」という敷居の高さがないところもいい。あとはPC-98のワクを抜け出してくれさえすればねぇ。

1994年は次世代ゲーム機が登場し，ゲームの流行も3Dモノに移りそうな気配だ。うかうかしてるとパソコンゲーム自体が危なくなっちゃいそう。ここはひとつ，「R.C.」みたいにAV機能と結びつくことで「非」アクションゲームにブレイクスルーが起きることを期待したい。

求ム，パソコンでしかできないヘンなゲーム！

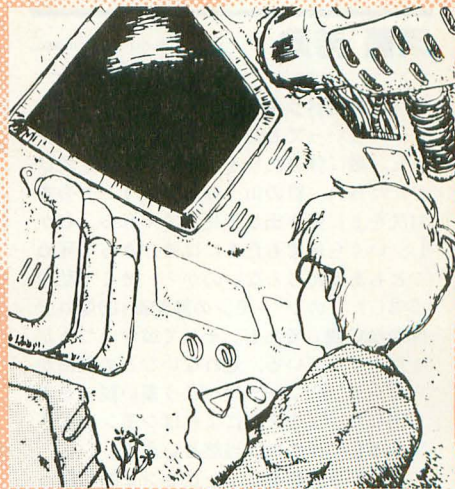
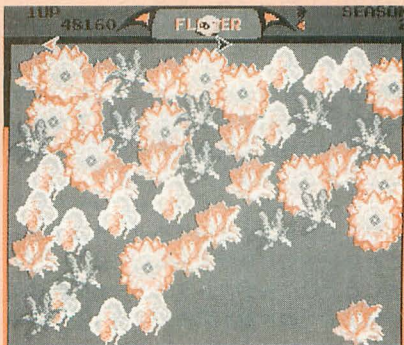
柴田 淳 Shibata Atsushi

1．ビデオゲームアンソロジーシリーズ
2．ロボットコンストラクションR.C.
3．GADGET
(Macintosh，FM TOWNS)

実は，電波マイコンソフトのなにわさんとは2～3度会ったことがある。僕が中学生のときで，彼がX1で「ゼビウス」を作った頃なので，あちらは覚えておられないと思う。ナムコの人に発売直前の「ゼビウス」を見せ，「マネはうまくなったな」といわれて悔しがっていたのが印象に残っている。

なにわさんが「リブルラブル」開発の折，オリジナルボードのロジックを解析したと聞いて，あのときの彼の悔しがっていた表情が思い出された。移植度の高さには定評のある電波だが，人間のこだわりとかいったものは，個人のブルーな部分に根ざしているような気がする。だから細部までキッチリ作られたものを見ると，僕は共感をおぼえてしまうのだ。そういった意味の1等賞である。なお，個別の作品を順位に挙げるとそれだけでベスト3が埋まってしまいそうなので，1位は「シリーズ」で，とさせていただいた。

キッチリ作られているという点では「R.C.」も負けてはいない。一見地味な印象のゲームだが，枝葉に至るまで妥協が見られない。たとえばロボットを作るパーツ選びである。脚になる部分に速いものを選ぶと燃費が悪くなる。こちらが立てばあちらが立たずという状況のなかで，最適な部品を組み合わせなければならない。「R.C.」には必勝法がないのだ。これはバランスが最高によく取れているからなのだが，必勝法がないからこそいつまでも遊べるのだ。これほどバランスの練られたゲームは，家庭用ゲーム機のソフトを探してもそうはない。

最後の「GADGET」はCD-ROMのゲーム。実は僕はこのゲームをやるまで「インタラクティブなゲームなんて，映画や小説にかなうわけがない」と思っていたのだ。CD-ROMという供給媒体を方法論から必然性に昇華させた，なんて書くとエラそうだが，場面の見せ方というか，並べ方が絶妙なのである。認識を改めさせられたゲームであった。アダルトものやとにかく詰め込むばかりのゲームの多い和製CD-ROMゲームのなかで，ひときわ光るゲームである。

illustration：Y. Kawahara

古村 聡 Komura Satoshi

1．極
2．餓狼伝説2
3．ストリートファイターⅡダッシュ

「餓狼伝説2」と「ストリートファイターⅡダッシュ」が移植されたのもビッグニュースだったけど，私にとって今年いちばんショックだったのは「コンピュータ将棋ソフト・極」の登場だったんだな。いままでの将棋ソフトってこっちが飛車角落ちでも「まだまだだね，ふっ」っていうレベルのものが多かったでしょ。だから「極」のレベルの高さには本当にびっくりしたんだよね。コンピュータ相手に指しても全然勝てなくて「ぅおのれ，たかが耐久消費財の分際で，この私を愚弄するのかぁ～っ！」って，何度こぶしをディスプレイにぶつけたことか……。おかげでディスプレイの上にはそのときついた血が乾いてできた黒いシミがついて，いまだに全然とれなかったりしてるんだな，これが。

ま，ともかくパソコン用の将棋ゲームで，平手でもまともに人間の相手になる将棋ソフトが出たってことは，たいしたことだと思います。こんなに強い＆それでいて速い思考ルーチンを作ったプログラマさんは偉いっ！

次点は「ライム」シリーズかな。

絵もキレイだし，バリバリアニメーションもするしでなかなかナイスだと思ったな。でも同時に，フロッピーディスクの限界を知ったゲームだったような気もする。実は「ライム」は1から5までをイッキに遊んだんだけど，通して遊んで分量的にちょうどいい感じがしたから，これって1本1本じゃちょっと物足りないんじゃないのかなぁ。やっぱり可搬大容量記憶媒体は必要かも。

ってーことで，CD-ROMでもいいけど，やっぱりOD-ROMとかパーシャルROMなら普通の媒体より安くできるっていうし，ソニーは安いドライブ出すらしいって話もあることですし，純正MOドライブ出しませんか？ シャープさん。ねぇ？

須藤 芳政　Sudo Yoshimasa

2．悪魔城ドラキュラ

なんといってもシモンの死に方がセクシーだ。敵に体当たりを食らい，火炎を浴びせかけられ，針の山へ転落しようとも必ずお尻を上に突き出して死んでくれる。しかし，いくら何でもたまには伸びきって死ぬこともあるんじゃないのか？　そこで私が提唱したいのが「シモンの膝関節100度以上開かない説」である。そして彼がいつも足を引きずっている，走れないなどの理由から「シモンのブーツけっこう重い説」を考え出した。この2説によればシモンが尻を突き出して死ぬのは当然といえる。

そして目の前をピョンピョンと跳ね回ってうざったいサル。私はあの頭のハゲ具合から「波平」と名づけた。試しにある大手かつらメーカーのテレビCMを参考にして，波平がワラワラと出てきたときに「それじゃあ私がかわいそう！」と叫んでみたが，まったく効果なし！ CMじゃ「ピタ！」っと動きが止まったのになー，残念。

初めて人形の館をプレイしたとき「お前は誰？」と聞かれて「シモンくんだよーん！」なんて飛びついて突き飛ばされたのは私だけではないことを信じている！

1．餓狼伝説

このゲーム，やはり気になるのはタン爺さん。試合中にもがきだすので，初めは塩辛の食いすぎで発作でも起こしてしまったのかと思ったら，いきなり巨大化！「うごー！」ぐりぐり回転して足から変なもの発射したりしてめっぽう強いが，しばらくすると元のちっこい爺さんに戻ってしまう。巨大化の時間が3分もたないのでウルトラマンの血を引いてはいないようだ。「餓狼伝説2」でステージ間のほんの一瞬だがダックキングと添い寝している現場をクラウザーにフォーカスされているところをみると，けっこう危ない趣味の爺さんかもしれない。みんな気をつけろ！

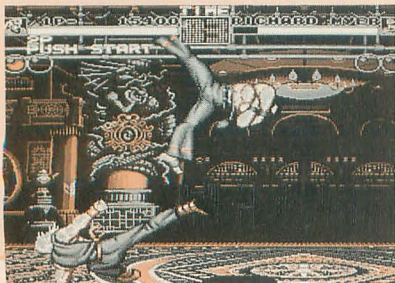
怪しいといえば，BGMからして怪しいリチャード・マイヤー。左右の腕を交互にブラブラ動かすさまは1970年代のディスコで踊るラッパズボンにダッサダサのトレーナースタイルでフィーバー(死語)している若者さながらだ。

「餓狼伝説あるよ」

「えーやらしてやらしてー！」

「ブーブーン……ブーブーン……」

ディスクアクセスの長さも，かなり怪しかった。

八重垣 那智　Yaegaki Nachi

1．悪魔城ドラキュラ
2．コットン
3．究極タイガー

いつもの調子で，勝手気ままに3本選んでみた。そもそもお気に入りになりそうな作品は，親の敵と思われようと万難を排してレビューを担当してしまうので，ベスト3が担当したソフトに偏るのは当然の帰結。あとは，今年いちばんの衝撃を受けた作品を押さえて一丁上がりである。

「ドラキュラ」が入っているのに「ストⅡダッシュ」がないのは変だと思う方もいるかもしれないので，一応フォローしておくと，これはどれだけ新鮮なインパクトがあったかの問題である。見慣れすぎてしまって食傷気味の格闘モノよりも，ファミコンの焼き直しという予想を遥かに凌駕したモノが凄く見えるのはあたりまえ，という感覚は私だけのものではないと主張しよう。それなら同じ忠実移植型の「究極タイガー」は何なんだという反論もあるかもしれないが，これはレビューにも書いたとおり，いちばん好きなゲームだから別格。ちょっとやそっと流行っているようなゲームでは，太刀打ちできるわけがないことは，皆さんにはわかってもらえるだろう。まぁ，これは完全に私の好みの問題だが，それですら3位にしかならないというのは，あくまでもインパクト重視で選んだということなので，どれだけ「ストⅡダッシュ」や「餓狼伝説」の出来がよくて私が実際にそれを認めていても，ここでは関係ないのである。

だから「コットン」のリニューアル移植は高く評価するし，「ドラキュラ」をトップに選ぶわけである。驚きは人間の感情としては，最も強く激しいもののひとつであり，それを揺さぶるゲームは心に残り，愛着が生まれてくるということに，なんら不自然なところはないだろう。

比較的X680x0のゲームの規範となっているアーケードゲームに一点集中傾向というか，ワンパターンな流れが飽きもせずに続いているので，いまにX680x0のゲームも，掃いて捨てたくなるのではないかと危惧していたが，今年もこうやって選ぶために久しぶりに振り返ってみると，まだまだ捨てたものではないことがわかる。

人それぞれがゲームに求めていることは違うというのは承知しているが，エンターテイメントというものは，常に新しい見たことのない何かを求め続けていくものである。映画など，いくらヒットしても何年もロードショー上映されたりはしない。たまに名画座でよさを再確認することはあっても，映画は常に先に進んでいるし，それを楽しむ人も進歩している。ゲームも，ゲームを遊ぶ人も，新しい驚きを求めて進歩することを忘れてはならない。それがなくなってしまうのは，ゲームの死を意味するからだ。

illustration:T.Takahashi

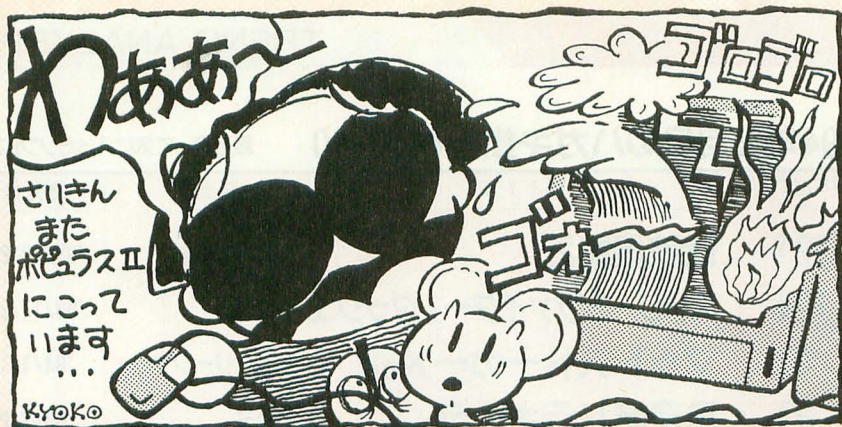
西川 善司　Nishikawa Zenji

シムアント
エトワールプリンセス
悪魔城ドラキュラ
ストリートファイターⅡダッシュ
餓狼伝説2

「シムアント」はよかった。SX用という点も評価高い。生態シミュレーション的な匂いを漂わせつつ，戦略的リアルタイムシミュレーションの面白さも兼ね備えるデザインの上手さと奥深さは感服せざるを得ない。

「エトワールプリンセス」も，キャラクターの魅力と世界観の演出に長けた作品であった。これは続編の可能性もあるのでは？　そのときにはぜひ，仲間キャラクターの同時使用や斜め方向への攻撃を可能にするなど不満点を解消しパワーアップしてほしい。

「悪魔城ドラキュラ」は，ずいぶんと夢中にさせてもらった作品。久々の「先を見たくなる」「何周もしたくなる」ゲームとして評価したい。高いといわれる難易度も何度も練習すれば進めるような絶妙なバランス設定だ。ステージクリア時の充足感は難易度が低かったら醸し出せなかったろう。結局この難易度でよかったと思う。また，X68000オリジナルという点も特筆に値する。

「ストリートファイターⅡダッシュ」はX68000ゲーム史上，忘れてはならない作品。完成度は高く，購入者の満足度は限りなく満点に近かっただろう。あえて不満点をいえばサウンド周り。ハードの限界を超えたポリフォニックAD PCMドライバは評価したいが，鳴っている音に納得がいかない。特に効果音などはアーケード版のファンを憤慨させた。各キャ

高橋 哲史 Takahashi Tetsushi

1．リブルラブル
2．エトワールプリンセス
3．スターフォース

期せずして電波新聞社のアンソロジーシリーズが2つも入ってしまいました。やっぱり自分がゲーセン小僧だったあの頃が深い影を落としているんですね。

「リブルラブル」は「ボタンがないヘンなゲーム」でしたが，ゲームの斬新さ，グラフィックの美しさ(のちに某誌に載っていた拡大ドット絵をX1でコピーして遊んだりもしましたっけ)にとりこになった覚えがあります。ああ，懐かしい……。

3位の「スターフォース」は実はちょっと事情が違って，ゲーセンではなくX68000で初めてプレイしたので，私にとってはアンソロジーでなく新作といえます。正直，最初は何の気なしに遊び始めたのですが，ずぶずぶとのめり込ませるゲームデザインに，シンプルイズベストの極みを見た思いがしました。結局かなりの時間をこいつに吸われてしまいました。

そして2位の「エトプリ」は久しぶりにX68000オリジナルで「面白い！」と思えた秀作です。しかも変に肩に力が入っていたりせず，楽しんで作ったという雰囲気がひしひしと伝わってきて，非常に好感のもてる仕上がりでした。個人的には「エトプリ」で「EXACTな色使い」も独自の完成を見せたと思っています。「ジオグラフシール」にも期待したいところです。

あと，この1年で記憶に残っているゲームといえば，やぱりMEGA-CDの「シルフィード」でしょうか。バックに流れるポリゴンな背景は，私の3Dスピリッツを熱く刺激してくれました。元祖「シルフィード」が発売されたとき「X1で出るまではやらん！」と意地をはったばかりにあまりプレイできなかったウサを存分に晴らさせていただきました。やっぱ男は黙って3Dだぜぃっ。

最後にもうひとつ気になるのが，私の仲間うちで妙に盛り上がっている「ロボットコンストラクションR.C.」です。発売当初から気にしてはいたのですが，ちょっと油断しているうちにみんなのロボットはめきめきと強くなってしまい，実に始めづらい状況になってしまいました。やはりこの手のゲームは仲間と同時期に盛り上がっていかないと辛いモノがありますよね。「COMSIGHT」や「CORE WARS」も結局やりどきを逃してしまったし……。しかしこれ以上R.C.人気が続くならば，私も黙ってはいられないと思っている今日この頃なのです。

ラクターボイスのピッチにオリジナルと異なり違和感があり，さらにガードしたときの音はパフという変な音で，手首に剃刀を当てた熱狂的ファンまで出現したとか。「スーパーストII」の移植の際には考慮してほしい。

最後は，「餓狼伝説2」。前作よりも格段に移植完成度は上がっている。ゲームの出来は素晴らしく，1993年最も夢中になったもののひとつだが，遊ぶにつれて不満点も見えてきた。まず，付属パッドのボタン配置が変。また，奥ラインでのキャラクターの表示がおかしい。小犬顔になるキム，1つ目妖怪になるチン……。体力ゲージが8ドット単位で減るのもなんか安っぽいイメージ。画面解像度もなるべくアーケードに合わせたものにしてほしい。あと，いまでは同人ソフトでさえやっている拡大縮小。これはぜひとも今後は導入してほしい(オリジナルのNEO・GEO版では試合後のボーナス換算時のメッセージやエンディングポーズは拡大処理されつつ表示される)。10MHz機に合わせた仕様だろうが，BGMのPCM音がごっそり削られているのも残念。「ジュジュジュジュジューベー」のラップサウンドや「ハヤハレハラホヒー」のタイ民族音楽フレーズが鳴らないのは悲しい。PCM8.Xの使用を前提のサウンドルーチンでもかまわないと思う。また，パッドよりも変換アダプタのほうが喜ばれたかも，というのが消費者の正直な意見。2ボタンスティックと専用パッドでは必殺技を出す操作感覚が著しく違うのも残念だ。魔法株式会社はおそらくX68000ユーザーの前に現れた新たなるホープである。今後もよい作品を送り出してほしい。

ところで，ゲーム発売タイトル数は減りつつある。新しい風を起こすためにもシャープのハード面でのサポートを期待したい。

横内 威至 Yokouchi Takeshi

そういえばもう1年たったんだよな。この1年といえばやっぱりアレか。トップはやっぱり「ストII」だろうな。思えば，絶対出るとは予想していてもなかなか発表なし。焦らされ抜いたあとの突然の発売。わかっていながらもやはりショッキングな1本であった。ただし，「ターボ」に慣れてしまった俺としては一部物足りなさを感じている。今年はさらに「ターボ」，「スーパー」あたりも期待している。

さて，次いではやはり「ドラキュラ」が熱かった。細部までこだわった綿密な構成，限界を知らぬ数々のエフェクト，この映像には本当にシビレさせてもらった。もとから「ドラキュラ」シリーズに思い入れがあるからこそ楽しめたのでもあるが，やはりあの統一されたムード作りはさすが。あえてオリジナルとしたのもコナミの良心。X68000であることを誇りに思わねば呪われる。

熱い，といえばいまは「リッジレーサー」。俺の右手親指の内側はもう皮膚が存在していない。命を削ってドリフトに励む。ここまでハイな車ゲーは未体験。もはやあの「バーチャレーシング」でさえ色褪せて見えてしまう。遂に新たな世代に到達しようというこのゲーム，バーチャルリアリティ時代の幕開けとなるであろう。無論完璧とはいわない。ソフトが進化すると不満もレベルは高まる。Gが感じられない，路面からの抵抗がまったくなくて車が軽い，コースが比較的単調，せめてフェイントモーションでクリアするコーナーが欲しいなど。明らかに贅沢な不満だ。また，「バーチャ」に比べ演出が寂しい。デモを見てもせわしなくてちょっとね。まあ，すぐに次がくるさ。

さらに今年はレベルが高かった。「餓狼2」もかなりいける。ただし個人的には「ストII」のほうが好き。次の「餓狼スペシャル」はかなり期待している。でも，「侍スピリッツ」が俺は欲しい。ということで魔法株式会社さんは期待No.1ソフトハウスだ。そしてアンソロジーシリーズも渋い。忘れ難い少年の日のあの1本が出てくればNo.1の可能性も当然ある。

全体として，レベルの高いユーザーの要求を満たせないソフトは淘汰され，一流の質を誇るゲームメーカーが残ったのが今年の動きだ。そういう意味では非常に好ましい状況だったのかもしれない。スーファミよりも遥かに楽しめるのだ。スーファミで面白いといえば「ストIIターボ」だけ。

これからはどのような動きがあるのだろう。格闘ゲーム時代へ移行するのか？　アクションが盛り返すか？　やっぱりピットファイターが俺の一生の願いだ。コナミさん，頼む！

ということで，

1．ストリートファイターIIダッシュ
2．悪魔城ドラキュラ
3．リッジレーサー

TREND ANALYSIS

1994年3月号のハガキ集計ベスト10

最近買って気に入ったソフトは？

POINT	タイトル	発売元	発売日
121	餓狼伝説2	魔法株式会社	'93/12/23
104	ストリートファイターⅡダッシュ	カプコン	'93/11/26
41	マッドストーカーX68	ファミリーソフト	'94/1/14
27	悪魔城ドラキュラ	コナミ	'93/7/23
22	卒業～GRADUATION	ブラザー工業	'94/1/29
19	ドラゴンバスター	電波新聞社	'93/12/10
14	SX-WINDOW ver.3.0	シャープ	'93/3/30
11	コットン	EAビクター	'93/9/24
11	MATIER ver.2.0	サンワード	'93/10/20
8	ぶたさん	電波新聞社	'93/10/29
8	キーパー	サクセス/電波新聞社	'93/12/22

（無作為抽出した1000通のハガキを集計）

2月号のハガキはGAME OF THE YEARの投票用紙となっていたため，先月号はこの集計はお休みでした。

つまり，前回のデータは2月号に掲載したものになるわけですが，読者のみなさんは覚えているでしょうか。そう，「ストリートファイターⅡダッシュ」が，それまでの3カ月の間，ポイント数も過去最高という圧倒的な強さでトップの座についていた「悪魔城ドラキュラ」を，あっさり抜いて1位に躍り出て，なおかつポイント数でも新記録を達成したのでした。

ところが……。盛者必衰は世のならわしとか。ひと月遅れで発売された「餓狼伝説2」に早くも今月トップの座を追われることになりました。とはいえ，獲得ポイント数は3桁の数字をキープ。ひところの集計ならば，1位間違いなしの支持を受けているのですから，ハイレベルな戦いになったものですね。

で，その1位の「餓狼伝説2」。1993年7月に発売された前作「餓狼伝説」は高ポイントを獲得しながらもドラキュラ人気に押されて結局1位の座につくことはできませんでした。その雪辱，というわけではありませんが，ゲームのパワーアップに加え，専用パッドつきという力の込めようで，なかなか高く評価されています。

3位「マッドストーカーX68」は初登場。X68000には久しくご無沙汰だったファミリーソフトの製品で，これも対戦格闘ゲームです。ロボットの動きの美しさは秀逸。

それにしても，同じタイプのゲームがベスト3を独占するなんて，めったにあることではありませんから，やはりこれは本当に大きな流行といわざるをえません。発売はそれぞれ1993年11月，12月，1994年1月と，みごとに月イチのペース。それでもどのソフトもそれぞれ強力なファンを獲得しているということは，やはりこのジャンルのソフト全体の水準の高さを示しているのでしょう。

魔法株式会社，ファミリーソフトともに次回作も対戦格闘ものを開発中とのことですから，このジャンルは今後さらなるレベルアップが期待されます。

そのほか初登場は5位の「卒業～GRADUATION」と10位「キーパー」。シミュレーションとパズルという，ジャンルとしては久しぶりのものです。アクションやシューティングばかりが注目されがちのX68000ゲーム界ですが，それ以外でも質の高いソフトはこれからもどんどん出てきてほしいものですね。

ゲーム以外では「SX-WINDOW ver.3.0」が人気を保っています。「EG Word」や「開発キット」の発売で来月のランキングへの影響はどうなるでしょうか。

[特集]

SX-WINDOWの活用

SX-WINDOWの登場から4年。五里霧中のなかでの最初の1年，大幅に高速化されたver.2.0が出て，もう1年，さらに高速化されツールも実用的な段階になって，もう1年。
そして，いま，ようやくユーザーの手にも正式にメーカー標準の開発ツールと日本語ワードプロセッサが登場しました。
開発ツールは，あまり新しい情報がないことと最新版のシステムには対応していないことなどの不満点はありますが，豊富なサンプルプログラムを収録しており，ようやく一般ユーザーでもウィンドウプログラムが作成できるような基盤を作ることになりそうな感じです。
またひとつの転機を迎えたSX-WINDOWシステム。
まもなくウィンドウ環境を標準とすることが当たり前の時代になるのではないでしょうか。

CONTENTS

マルチウィンドウのユーザーインタフェイス

人間不親切起源論

Nakano Shuichi 中野 修一

S X - W I N D O W

SXのGUI関連はMacintoshの影響が強く出ています。しかしMacintoshから受け継いだもののすべてがよいものというわけではありません。多くの選択肢のなかから，どういう考え方で使うかというところが肝心でしょう。

SX-WINDOWver.3.0が発表されてから1年。その後SX-BASICが発表され，SX-WINDOW開発キットもようやく発売されそうな気配になってきました。これらによって，エンドユーザーでもウィンドウプログラムが作れるような環境ができあがりつつあります。私たちも，もう少し真剣にユーザーインタフェイスについて考えるときなのかもしれません。

最初のSX-WINDOWから4年の月日をかけて，ようやく発表されたメーカーからのプログラミングガイドラインを見るといろいろと美しい言葉が並んでいます。

推奨されているもの，禁止されているもの，それぞれで違和感を感じるものもいくつかあります。

LOOK&FEELを統一するなら美しく統一することがもっとも重要です。操作性はその次でかまいません。SX-WINDOWはバージョンアップを重ねるにしたがって操作の美しさが損なわれてきたような気がします。もっともいけないのは意味のない統一をしてしまうことでしょう。不便さに慣れるというのはそれほど悪いことではないのですが，美しくないものに慣れるのは明らかに悪いことです。

GUIはわかりにくい

次に「GUIは初心者にはわかりにくい」という基本的な事実を忘れてはいけません。

ビジュアルであることとグラフィカルであることは違います。わかりやすく作るためのひとつの方法が「ビジュアルにすること」であり，「グラフィカルにすること」はさらにそのうちのひとつの方法にすぎません。

操作系の説明についてはグラフィカルであることが障害になることが多々あります。アイコン表示だけのグラフィックツールを見ると意外とわかりにくいものです。これは，見て即座に意味がわかるようなシンボルが限られていることに起因します。

かといって，勝手にボキャブラリを増やせるものでもありません。アイコンは見る人に共通認識があって初めて意味をなすものです。アイコン操作が有効なのは低機能であることが絶対条件だといえるでしょう。機能が増えていくにつれて，かえって邪魔でわかりにくくなっていきます。

ましてや，抽象形だけで機能を説明しようとするのは暴力以外のなにものでもありません。

たとえば「丸は日本語モードで四角は英字モードだ。どうだわかりやすいだろう」とかいわれても困ってしまいます。たまには脳味噌の左側を使うことだって大切だというよい例です。

絵自体は玉虫色の存在ですから，共通認識のとれるもの以外は使ってはいけません。ちょっと話は変わりますが，そういう意味では「四角のなかにバッテンマークをつける」ことが「その機能の選択」になるというのは日本人の感覚ではありえません。

逆に，せっかくの直感的に理解できるシンボルを使っていながらそれを否定し，印刷メニューで字詰めを決めるとか，ファイルメニューで印刷するといったような構成はまったく馬鹿げているといえます。

難しい動作とは

ウィンドウ環境ではたいていマウスを使います。同じマウスを使うシステムでも扱いやすさはさまざまです。マウスを使う際の難しい動作とはなにかを見てみましょう。次の各群を比べてください。

A　上下に動かす
　　左右に動かす
B　右に動かす
　　左に動かす
C　上に動かす
　　下に動かす
D　右クリック
　　左クリック
E　ボタンを押したまま動かす
　　ボタンを離して動かす
F　クリック
　　ダブルクリック
G　機能を選んで位置を決める
　　位置を決めて機能を選ぶ

人によって多少違いはあるでしょうが，私の場合，マウスは左右に動かすほうが楽ですし左よりは右に動かすほうが楽です。上下ではちょっと考え込む人も多いでしょうが，グラフィックツールで縦線を描くとき下から上に描く人は多分少ないでしょう。ただし，距離が長いと下に動かすのはつらいところですけど。

次にクリックです。

たいていの場合，左ボタンは人差し指の下，右ボタンは中指の下にあります。人によっては親指と薬指という場合もあるかもしれません（X68000以外ではありえないだろうなぁ）。いずれにしても左クリックのほうが楽に決まっています。E，Fについては考えるまでもないでしょう。

たとえば，シャーペンで文字の大きさを変えたいというときに私のマウスカーソルは，斜め上に走りアイコンの下を行き過ぎてから右横に戻ります。

水平方向の位置あわせはうまくできても垂直方向にはあわせにくいことが原因のようです。いろいろやってみると垂直方向にはわりとアバウトな動きですが，水平方向にはそれなりに正確な動きができます。

これをもとに機能メニューなどをどこに配置すればいいかを考えてみましょう。

機能選択などは作業エリアの右側でやるのがもっとも適していることが推測できます。ストロークが短ければ上下でもいいでしょう。この場合，上下の位置あわせは難しいので，ボタンなどには縦方向にある程

度の大きさがないとダメです。Macintoshのメニューはアバウトな操作でも確実に高さをあわせられますが，ウィンドウごとにツールバーを持っているWINDOWSやSX-WINDOWの一部のソフトではそうはいきません。

なお，SX-WINDOWver.2.0以降，デフォルト状態ではウィンドウの左上揃えにシステムアイコンが配置されていますが，私の使うシステムでは右上揃えになっています。理由はいうまでもないでしょう。

さて，最後に残ったGについて考えてみましょう。ちょっとわかりにくい表現ですが，なにを意味しているかというと，要するにカット＆ペーストの方法についての根本的な問題なのです。

ED.XやSX-WINDOWの操作では範囲の先指定，日本を代表するワープロでは通常，後指定となっています。それぞれ，

●**先指定**

先頭位置を指定
終了位置を指定
削除を指定　　　→　削除動作

●**後指定**

削除を指定
先頭位置を指定
終了位置を指定　→　削除動作

のように動作します。

どちらが楽かは人によって違うでしょうが，私は先指定のほうがわかりやすいと思っています。

ひとつの理由は，範囲を指定するほうが複雑な動作だからです。失敗する可能性のあることは先に行っておいたほうがいいという考え方です。もうひとつは動詞的意味の操作と実際の処理実行は対応させたほうがいいという感覚的な意味あいからです。

一般的にいって，このように操作体系を感覚的にしていくことがわかりやすいユーザーインタフェイス作りの基本となります（徳島あたりの人は違う意見かもしれないが)。

メタファーと操作性

「メタファー」，暗喩といった意味ですが，ここでは実際に私たちが行っている日常の動作にたとえた処理にすることだと思ってください。

ファイルを消すにしても「二等辺三角形の頂点にアイコンを置く」といった無機的なものより「ごみ箱に入れる」といったもののほうがわかりやすいということです。

スライドボリュームなどはその特徴を濃厚に持っていますね。

アイコンのドラッグというのは放り込むという行為の優れたメタファーです。また，階層化メニューは整理されてわかりやすい構造を持っています。しかし，そこで使われているマウスのドラッグという行為はあまり楽ではありません。というか，間違いの起きやすいオペレーションなのです。

このように「わかりやすい」と「使いやすい」は共存しないこともあります。

ドラッグ＆ドロップの場合はそれ以外の操作方法も用意されていることが多いのでそれほど問題はありませんが，メニューはできるだけ短く，間違って指定されてもかまわないものを登録するようにしたほうがよいでしょう。よく使うものを上にするのは当然ですね。

隠されたもの

そもそも，メニューというのは目に見えないユーザーインタフェイスです。Macintoshでは明示的になっていましたが，SX-WINDOWでは隠された機能がコントロールやシフトを押しながらメニューを出したりすることで現れることもあります。

これはインタフェイスをビジュアルにする方向とは逆のベクトルを持っています。

メニューというのは基本的に隠された機能です。それでも非常にわかりやすいユーザーインタフェイスであったのは，ひとえにそれが「操作の対象となるオブジェクトの位置で機能を選択する」というふうに使われていたからでしょう。

最近は操作オブジェクトとは関連のないところでメニューを開かせるようなインタフェイスがむしろ「推奨」されています。これはどうしても納得できません。

そのほか，機能が隠されている例として，見た目でわからない部分での操作性が違っていると最悪です。同じウィンドウアイテムなのに使い方が違うとか，違いが目に見えるはずのない日本語入力の方法などが違っていたら混乱は避けられません。

機能をみつけやすく，操作しやすくするためには，ボタンなどは必要なだけ大きくわかりやすくすることが必要なこともあります。

逆にウィンドウ自体は無駄なく十分に小さいことも必要です。なにしろマルチウィンドウ環境ですので，多くのウィンドウを並列して開いたほうが作業しやすくなります。あるウィンドウが大きければ，ほかのウィンドウを隠す可能性がきわめて高いのです。ウィンドウが大きすぎるために使い勝手が悪くなったアプリケーションもありますので，画面が狭いという基本的事実を忘れてはいけません。

情報の不足

すでにSX-BASICの基本構想が発表されてからずいぶんたちます。理想のマルチウィンドウ環境の模索から始まったタスク間通信によるアプリケーションネットワーク構想が，当時ようやく動き始めたSX-BASICを核言語として据えることにより大きく前進していきました。

SX-BASICそのものは当時現れてきたVisualBASICに大きく影響を受けています。ユーザーインタフェイス部分はそれで進化したといっていいでしょう。逆に立ち後れているのがアプリケーションインタフェイスです。

SX-BASICは単に，簡単にウィンドウを使ったプログラムを作るための言語ではありません。SX-BASICそのものは，ウィンドウを開くことはおろか文字表示くらいしかできません。ウィンドウエンジンというアプリケーション（一応汎用）の機能をあたかも自分の機能のように扱えるだけの話です。

現在かたちになっているのは，SX-BASICとウィンドウエンジンの連携だけですが，最終目標はすべてのアプリケーション間でこのように密接な連携を築くことにあります。CALCエンジンやグラフィックエンジン，ミュージックエンジン，データベースエンジンなどといったものの機能が自由に使えれば環境も向上するでしょう。

このようにSX-BASICシステムはアプリケーションによって支えられるものですが，最大の問題はアプリケーションが作りにくいということです。まともにアプリケーションを開発できる環境が与えられていなかったり，どんどん新しいマネージャが加わったりしています。

シャーペン．Xは柔軟でかつ高機能，拡張性も備えているのですが，拡張方法がわかりませんので，システムの核となるエディタにはなりそうもありません。となればエディタから作り直す必要があります。

今回，田村氏によって拡張テキストフォーマットの解析は行われましたが，まだまだ情報は不足しています。フォント構造やグラフィック表示ほかver.3.0の機能などは，できれば今世紀中には公開していただきたいものです。

SX標準ドローイングツール

Easydrawで回路CAD

Taki Yasushi 瀧 康史

S X - W I N D O W

発売以来，ユーザーから高い評価を得ているEasydrawSX-68K。綺麗な出力が得られるので図版作成に威力を発揮しています。ここではEasydrawを使った回路図作成と基本的な使い方について見てみましょう。

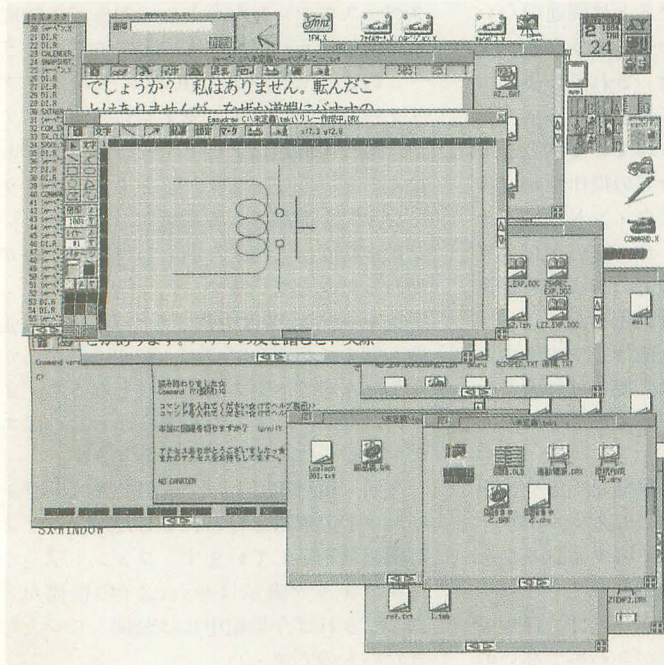
Easydrawの背景

最初にはっきりいってしまおう。

私は，まだEasydrawを十分には使いこなしていない。それじゃあ，記事なんて書くなぁっていわれそうだが，周りの皆さんがあまりにもEasydrawを知らないので，ちょっと，ペンをとらせていただくことに決めた。

つまり，Easydrawはまだまだ奥が深い。

私も結構使い込んだがまだまだいけそうな気がする。だから世の中には私以上にEasydrawを使いこなしている人はいっぱいいることと思う。ひょっとしたら，もう，私と同じような使い方をしている人もいるかもしれない。

要はこのソフトがSX-WINDOWで動いていることに意味がある。SXじゃなければCANVAS PRO-68Kがあった。でも，SXじゃないから使わなかったんだよな。

SXだから，私はこうやっていま，シャーペンで記事を書きながら，Easydrawで仕事できるし，SXだからそうしながら，通信ソフトを起動して，PC-VANのOLT(チャット)で5時間も6時間も惚けて仕事が……（ぬぅ，あんまりよくないじゃないか)。

つまり，SXの環境はまだまだといいたいけど，かなり成熟してきていることも確かだ。それは10MHzでは速度的に不満が残るが，マルチで動く便利さがそれを補ってくれる。

正直いえば，ファイルメンテナンスなんかは確かにコマンドシェル上からDIなどのメンテツールを作ったほうが使いやすい。

しかし，そういう使い方と，SXでのマルチタスクを生かした使い方はもはや別次元に達している。GUIのウィンドウマルチタスクだから初心者には使いやすいとか，どっかの雑誌が間違えて伝えた偽りは捨てるべき。キーボードのタイプが速いから，コマンドシェルでいいや……なんて，見当違いな公然の嘘は聞くべきでない。そんなのは，X68000ユーザーはみんなゲーマーといっているのと同じくらいにナンセンスな話。まぁ私はゲーマーだけどさ。しない人もいるでしょうに。

図1 SX-WINDOWの作業環境例

マルチウィンドウマルチタスクのシェルを使うには，人間がマルチタスクな使い方を覚えなくてはいけない。これは，使っているうちにわかることで，一朝一夕にわかることではない。悟ることかもしれない。

SXには使いたいソフトがないとか，ひとつのアプリケーションを使っている途中に，ほかのソフトを使わないだとか，使っても子プロセスで実行すればいいなんていい方は，お門違いも甚だしい。

要は，アプリケーションを実行中に，「いつでもほかのソフト」に移れ，ほかのソフトから「なんでも」持ってこれ，それを合成して作業を行えば，どのような「可能性」を見い出すことができるか。「できる」か？というのは問題ではない。なにが「できあがる」のか？　というのが問題。

そういう背景のもとに，こういうドローイング系のツールがある。どのような価値があるかな？

それでは，Easydrawに具体的に触ってみることにしよう。

ライブラリを揃えなくっちゃ

Easydrawはお値段は安いけど，いろんな機能を持っている。なんにしろ，結構使いやすい。

Easyというのは，簡単に「作った」ソフトでもないし，簡単な「出来」のソフトじゃない。簡単に「扱える」というのが前提である（と思う）のを忘れずに。

しかし，反面，こんな出来のいいソフトが19,800円だということのオチのように，ライブラリがなにせ少ない。もっとも，一般的な人間が使いそうなライブラリばっかといえばそうなのだけど。多分，年賀状とかのポストカードや，ポスターなんかを作るためのライブラリなんだろうな。

とりあえず，なにをするにもライブラリを登録しなくっちゃ話が進まない。ドローイングツールはなんでもそうかもしれないけどさ。

とりあえず最近私はわりとよく電子回路を描くのでEasydrawを回路CADに化けさせてみた。

まずもっともよく使う部品，「抵抗」を作ることにした。いま考えるとバカげた話だが，最初私は実寸大で描いてしまった。よくよく考えてみれば，ドローイングツールなんだから，でっかく描いておけばいい。

適当な大きさでグリッドをうまく生かして，線をつないで大きく抵抗を描いてみる。多角形線を使って最後に閉じずに終了してもよい。座標は適当。私は印刷されるときに線が0.5mmぐらいの太さになってほしかったので，この上でも0.5mmにしてみた。

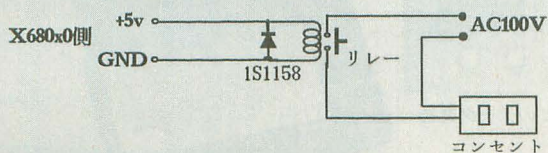
そのあと，それをマークし，メニューを出してグループ化。あとはリサイズをすれば十分な大きさにしてしまえる。各自適当な大きさにしておくのがいいだろう。

その状態でまたマークして，ライブラリ編集を呼べば，ライブラリは登録される。もちろん，ライブラリ名は，適当にリネームしておかなくてはいけない。最初の状態だと，Easydraw.dlbになってしまってるから（私は回路.dlbにした）。

同様にしてダイオードも作ってみた。ペイント系ツールじゃないから縮尺自由なのがよい。ダイオードの三角形は，最初にパターンを黒塗りに設定し，多角形アイコンで，三角形を作って結線終了する。縦棒は，1mmぐらいで描く。あらかじめ大きく描いているときは，縦棒の1mmは細すぎるように見えるけど，小さくするときのことを考えて，適当にしておくように。

なぜならあくまでもドローイングツールだからね。線の太さはあとから代えられるけどライブラリ登録してしまうとできないっぽいし。

割合難しいのが，コイルなどの螺旋状の図。しかし，グリッドをうまく利用して大きく描き，最後にまとめて縮小する。（結構時間がかかってしまうが）ベジエ曲線やスムージング多角線を利用してもよい。

という具合に，ライブラリをサクサク作成していくとよい。最初にまとめて根性で作ってもよいが，使うたびにちょくちょく作ったほうが楽かもしれない。

また，ライブラリ登録した図も貼るときに拡大縮小できるので，自分が使いやすい大きさで保存しておき，貼るときに伸長するとよいだろう。

拡縮自在できれいな印刷

シャーペンで印刷したことがある人はわかると思うが，実際の画面の大きさに比べてプリンタの解像度が高すぎるために，予想どおりの大きさにはなかなか出力できない。はっきりいって，慣れを必要とする。

ドローイング系のツールは基本的に，グラフィックのイメージをビットパターンとして記録しておくのではなく，座標だけを記録しておく。

確かにペイント系のグラフィックツールのほうが，見かけの2次元的なグラフィックが綺麗に描けるが，これでは実際回路を拡大縮小したときに問題が起きる。縮小ならまだいいけど。

これがドローイング系のツールであれば，ファイルフォーマットさえあれば，どのような解像度の端末に移動しても，適当に見やすい大きさで見ることができる。拡大縮小も自由自在。標準的なプリンタは最近では，360dpiか300dpi。対して画面はインチ数によって違うけど118dpiぐらいかな？

そういうわけで，画面上では16×16ドット文字で見やすいかもしれないが，360dpiのプリンタでは新聞の文字よりも小さくなってしまう。それを防ぐために，シャーペンでは，フォントをすべてベクトルフォントにすると，大きくなれば大きくなった分だけ綺麗に印刷されるようになる。

まぁ，180dpiの一世代前のプリンタ所有者の私には24ドットで最初から打っていれば問題ないけど。それでも，180dpiで作った資産が，360dpiのプリンタをあとで買ってもそのまま生かせるってのはおいしいよね。

Easydrawではイメージの貼り込みとか当然できるけど，拡大したときにデジタル拡大になってしまって，汚くなってしまう。だから，できる限りそれはしないほうがいい。文字もフォントはベクトルフォントを使う。

ドットが細かいので，細かい部分はなかなかうまく描けないけど，それは慣れ。特に端子の○なんて，小さいとなかなかうまくいかない。そんなときは全体を拡大してみてエディットしてみる。そんな手順で回路を描いていく。

とりあえず，簡単な回路を作ってみた。

まぁたいした図ではないが，Easydrawで回路図が割と簡単に描けるというのがわかるだろう。

図2

IO-735で出力（180dpi）

まとめ

線は簡単に結べるし，拡縮自由自在。それでいて，好きなときに好きなところを修正できる手軽さがいい。適当なときに，サクってウィンドウを開いて，思ったときに好きな回路を作れる。ライブラリさえあれば，楽譜も描けるかもしれないな（無理があるかぁ？ んん？ 無理じゃないかもしれないぞ）。

もちろん，クリップボードを通して，シャーペンの上に持っていけるところもいい。まぁ，文章を書く機能も充実しているけどね。わりと。あとは，印刷機能が充実したワープロソフトがほしいところ。もちろんベクトルフォントを利用して，どんな解像度でも同じ大きさにプリントアウトできるもの。

それに，当然Easydrawそのものを，メモリのある限りたくさん開けるのも素晴らしい。

欠点といえば，ライブラリを取り出すのが少々ややこしいことだろうか。回路みたいにライブラリ命なものを描いてると，opt.1＋Lを押しまくってしまうんだな。次のバージョンでは，疑似ダイアログなんてやめて，ほかのEasydrawからも引っ張ってこれるような，別ウィンドウにしてほしいところだ。

あとは，矢印の種類をエディットできるといいな。回路を書いてると線の先に○がほしいから。

それから，メモリブロックをたくさん残して，終了してしまうのもやめてほしいなぁ。ちゃんと解放してほしい。そういう仕様なのかもしれないけど。複数のウィンドウを開くときに，メモリブロックを共有化してるとか？（解析なんてしてないからわからんけど）

あぁ。そういえば来月号は付録ディスク号だったなぁ。ちょっと大きいから入らないかもしれないけど，スペースが「余った」なら，私が今回使った回路用のライブラリを入れてもらうことにしよう。

余談だが，ローテク連載の今月の回路ではEasydrawは使っていない。

新製品

DoubleBookin'

Tan Akihiko 丹 明彦

S X - W I N D O W

DoubleBookin'は，SX-WINDOWで動作するスケジュール管理ソフトウェアです。毎日のスケジュール管理はもちろん，周辺機器の制御や電子手帳とのデータ交換など，さまざまな拡張機能をもっています。

できること

DoubleBookin'は，1日単位の予定はむろんのこと，複数日にまたがる長期の予定や，毎週何曜日，毎月何日といったパターンの予定を管理できるようになっている。1日のスケジュールも分単位で登録することができる。

また，スケジュールをイベントとして登録しておけば，その時間にパソコンに何かをさせることができる。指定の時刻になったらベルを鳴らしたり，音楽を演奏したり，テレビのチャンネルを変えたりといった動作が可能になっている。このひとつとして，MIC-68K(アンフィニーシステム社製の赤外線による家電製品コントロールユニット)を制御する機能がある。活用次第では，いろいろな電化製品を作動させたり，停止させたりなどが実現できるのだ。

また，予定の多角的な表示が可能である。カレンダーウィンドウでは1カ月以上の範囲を見渡すことができ，予定が登録されている日にはマークがつく。デイリーウィンドウではその1日の予定を詳しく表示することができる。また，グラフウィンドウでは何時から何時までといった予定を視覚的に表示することもできる。

X68000用　3.5/5"2HD版　12,800円(税別)
計測技研　☎0286(22)9811

インストール

基本的な機能を理解したところで，使用してみよう。まずインストールはきわめて容易。パッケージに入っているフロッピーディスクからハードディスク(でなくてもいいが，快適に使うためには高速かつ大容量のメディアが好ましいことはいうまでもない)の適当な場所にプログラムディレクトリをコピーするだけ。あとは，DoubleBookin'のアイコンを登録するために同ディレクトリ内にあるDBIcon.Xのアイコンをダブルクリックすると，DoubleBookin.Xのアイコンが専用のものに変わる。

ユーザー登録

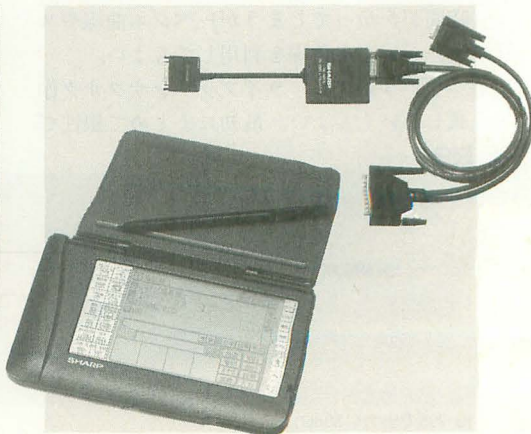
DoubleBookin'では複数のユーザーに対応しており，各人ごとのスケジュールを管理できる。このへんが基本的に個人向けのシステムである電子手帳とは異なるところだろう。

DoubleBookin.Xを起動すると，ユーザーウィンドウが開くので，ここでユーザー登録を行う。ユーザーの名前と属するグループ名，情報を格納するディレクトリ名を指定すると新規ユーザーが作成される。ユーザーはユーザーウィンドウに顔のアイコンとして現れる。以後はその顔をダブルクリックすれば，その人のスケジュールの参照や編集ができる。顔のアイコンは数種類が用意されており，パターンエディタなどで描いてカット&ペーストすればオリジナルの顔を登録することもできる。

しかしこういってはなんだが，ユーザーを複数登録できるメリットは現実にどれほどあるのかはっきりしない。パソコン特にX68000を何人かで共同使用しているユーザーというのは少ないように思うのだが。まあ，使い方によっては便利だし，家庭や企業での使用も考えられるので，機能自体に問題はない。ただ，複数ユーザーに対応するのなら，パスワードが設定できるようにしてほしかった。このへんは考え方次第ではあるが，仮にも個人情報である。他人に自分の手帳をのぞかれるというのは気持のよいものではないだろう。

スケジュールの実際

ユーザーウィンドウから顔をダブルクリックするとその人の情報にアクセスできる。具体的にはカレンダーウィンドウが開き，今日の日付が選択されている。ポップアップメニューでデイリーウィンドウを開くよ

RS-232Cを介して，電子手帳などとのデータ交換ができる
写真右上：レベルコンバータ
写真右下：シャープのPI-3000(ザウルス)

う指示すると，今日の予定にアクセスできる。別の日をカレンダーから選んでデイリーウィンドウを開くと，その日の予定にアクセスできる。デイリーウィンドウへの予定の登録は，予定の新規作成によって行う。何度も行えば予定は追加される。削除や修正も可能である。要するにウィンドウのなかでマウスをあれこれ動かしていれば操作は覚えられる。通常のSX-WINDOWアプリケーションのノリで扱えるのだ。

予定の1つひとつには，予定内容だけではなく，必要なら時刻，時間帯，イベントを設定できる。たとえば，見たいテレビ番組の時間とチャンネルを設定しておけば，開始時刻にパソコン画面をテレビに切り替え，さらに番組終了後に自動的にパソコン画面に戻る，などという使い方ができるだろう。基本的にはX68000でできることをなんでも設定できるので，開始5分前には予告音楽を演奏するとか，起動しているアプリケーションを終了させておく，などというきめ細かな動作も可能である。

ひとつ残念なのは，検索機能がやや弱いことである。もちろん，普通の検索なら問題はないのだが，一歩進んだスケジュール管理を考えるならば，複数の予定を同時検索して一覧表示する程度の機能はほしいところである。人にもよるだろうが，だいたいにおいてスケジュールというのは同じ項目が何回もある場合が多い。ゼミの出席予定とか，デートの予定一覧などが即座に表示できたら，スケジュール調整にも力を発揮するだろう。

カレンダーウィンドウからポップアップメニューでデイリーウィンドウを開くと予定が表示される

登録されているスケジュール一覧を見たり，グラフ表示したりもできる

印刷

作成した予定は，月単位，週単位，日単位で印刷できる。システム手帳のリフィルに使える形式で印刷されるので，枠線に沿って切り取って穴をあければ使える。リフィルの右ページに使うか左ページに使うかで穴のあけ方を変えることもできるし，A4のプリンタ用紙1枚に2ページまで印刷できるので折って使うこともできる。

電子手帳とのやり取り

DoubleBookin'はRS-232Cを介して電子手帳などとのデータ交換ができる。DoubleBookin'で入力した予定を電子手帳に転送することやその逆が可能。

対応している電子手帳は，シャープ製で「オプションポート15」を装備しているもの。具体的にはDB-Zシリーズ，PV-F1，PI-3000(ザウルス)である。これらの電子手帳とX68000/030を接続する際には，シャープ製のレベルコンバータと呼ばれるケーブルを使用する。

今回はザウルスを借りてきてデータ転送を試してみた。近頃話題の情報携帯端末というやつだ。電子手帳にペン入力がついたものと思えばいい。ザウルス側の設定にちょっと手間取ったが，問題なく転送できた。ということが確認できたので，ザウルスでひとしきり遊んでしまった。ペンによる文字の手書き入力がけっこう面白い。これが意外によく認識する。

しかし，認識はするのだが，やはり漢字の入力は手間がかかる。少なくともキーボードで苦もなく日本語を入力できるタイプの人間にとってはかったるい。ここにDoubleBookin'のようなソフトが大きな価値をもつのだろう。スケジュールを管理する機器は持ち歩けないと困るし，かといって携帯端末は大量の入力に向かないからだ。

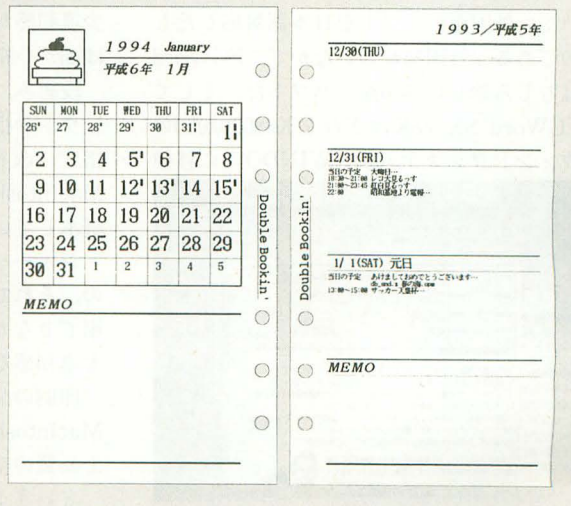
ひとつ注意しなければならないのは，データ転送は，最新データを書き加える形になることである。データ全体の整合性をとるわけではないので，データの追加については問題はないが，変更には完全には対応していない。たとえば待ち合わせ時刻を変更した場合に，変更後のデータは追加データとして登録されるが，変更前のデータは何もしなければ削除されずにそのまま残ってしまう。パソコンと情報端末の両方を併用して厳密なスケジュール管理を行うには，まだまだ人間自らがこまめに両方をチェックする必要があるようだ。

しかし，ともあれこの転送機能のおかげで重複入力の手間はだいぶ減ることになる。データが一元化できるという点は，非常に重要なことである。

ただし，現在の仕様ではほかの形式で利用することはできない。パソコンでのデータ処理を考えると，テキストファイル形式に落とせないのは残念である。

拡張性

DoubleBookin'は拡張性の高さをそのウリのひとつとしている。スケジュールイベントのためのモジュールをあとから作成して追加することが可能なのである。したがって，これまで述べたような不満点の解消や，将来出てくるであろう新たな機能拡張の要望へも，対応できる可能性がある。

ユーザーの要望によっては，今後も拡張モジュールを開発・発売するということであるが，それだけではなく，むしろ技術情報を公開してユーザー自身が拡張モジュールをプログラムすることが推奨されている。そのためのヘッダファイルも添付されていて，この部分については著作権はフリーである。ユーザー各自が，自分に便利な仕様にできるわけで，このオープンな姿勢については評価したい。

システム手帳のリフィル形式で印刷

EGWord SX-68K

新しいSX日本語環境を試用する

Tan Akihiko 丹 明彦

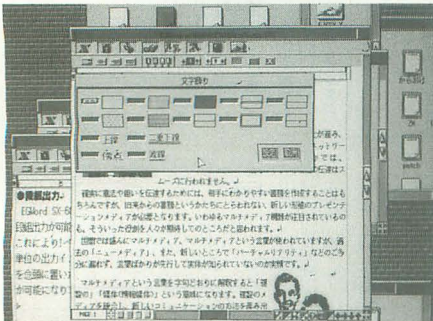
S X - W I N D O W

ついに姿を見せたSX-WINDOW用日本語ワードプロセッサEGWord。新しい日本語変換システムEG-Convertを搭載し，SX-WINDOWの機能をフルに活かしたヒューマンワープロの登場です。

ついに登場!

待望久しかった日本語ワードプロセッサがついにSX-WINDOW上に移植された。Macintoshの世界で有名なエルゴソフトの「EGWord」を，ほとんどそのままのかたちでX68000/030で利用することができるのだ。SX-WINDOWも日本語作業環境としてようやくひととおりのものを揃えたといえるだろう。

これまでSX-WINDOWには，ペイントツール「Easy paint」，ドローツール「Easy draw」など，単語の頭に「Easy」をくっつけた名前のソフトがあった。このワードプロセッサは「EGWord」である。まぎらわしいが両者はまったく別の生い立ちを持つソフトである。ちなみに東芝のノートブックパソコンには「ダイナブックEZ」というのがある。関係ないか。

EGWordは，かのMacintosh上の日本語ワードプロセッサである。MacWriteに日本語版のマックライトがあるように，またMacDrawに日本語版のマックドローがあるように，Macintoshの著名なソフトはたいてい外国産のソフトを日本語対応したものである。外国勢が多いなかで，EGWordはむしろ珍しい純国産のソフトだ。そしてEGWord SX-68Kはそれを X68000/030のウィンドウシステムSX-WINDOWに移植したものである。

装飾の例

インストールと起動

インストールは専用のインストーラによって行う。指示に従うだけでよい。なおEGWordはハードディスクでしか使うことができない。

起動はEGWord.Xのアイコンをダブルクリックするだけ。

起動して出現したウィンドウを見た第一印象は「Macintosh版そっくり」である。ウィンドウ最上段はSX-WINDOW風のプルダウンメニューだしスクロールバーもSX-WINDOW風だが，それ以外はMacintosh版の少し昔のバージョンと同じである。Macintosh版のEGWordはバージョン5が最新だが，EGWord SX-68Kはそれ以前のバージョンをベースにしているようだ。

文書作成ツールとしてのEGWord

EGWordは競争の激しいMacintosh出身だけあって，とりあえずマニュアルなしでそこそこ使えるし，SX-WINDOWでは多少違和感があるが，あちこちつついていれば細かい指定のしかたも覚えていく。

段組み，ヘッダ，フッタなど文章のレイアウトの指定，文字のフォントやサイズの指定，改行幅の指定，罫線の処理など，Macintosh版が備えていた機能をそのまま継承している。

シャーペン.Xでは裏技的にできたものの，それでも十分に拡大しないと綺麗に印刷できなかったEasydrawからの貼り込みも違和感なく行える。

印刷の品質もそれなりだが，残念ながらMacintosh版より劣る。フォントの豊富さと品質の格差はいかんともしがたい。

ということで，ワードプロセッサとしてのEGWordは及第点か，それ以上の評価はしてもいいと思う。ひととおりの機能は備えているし，DOSの文化をひきずったワープロよりはずっと使いやすい。

文章入力ツールとしてのEGWord

と，ここまでは誉めているように見える。しかし，いざ使ってみた場合にひと言ふた言いいたくなってしまう。なぜか。EGWord SX-68Kはテキストエディタとしての基本がなっていないからである。

とりあえず許しがたいのはキーバッファがたまること。カーソルをキーリピートつきで移動すると，指を離したあともカーソルが動き続ける。カーソル移動程度についてこれないというのは不思議。グラフィック処理が遅いのはわかるが，文字だけのときも遅れる。これではあんまりではないか。

さらに論外なのはデリートキーもたまってしまうこと。これではテキストエディタとしての基本がなっていないといわれてもしかたがあるまい。ちなみにMacintosh版ではキーバッファはきちんと処理されている。

あと，全体に処理がもたついているのも気になる。気になるというのはX68030でやっていての話だ。文字の挿入やページ移動，ウィンドウのアップデートといったごく基本的な動作でもマウスカーソルが踏切に変わり，ほんのちょっとだが待たされる。右ボタンを押してメニューが開くのにもひと呼吸おいた感じだ。

ほかにもメニュー類の挙動が妙だったり，やや不安定だったりと，全体にSX-WINDOWでのプログラミングに慣れていないという印象を受ける。それともMacintosh版とほとんど同じウィンドウ構成にしたのが余分な負担になっているのだろうか。

日本語入力システムEGConvert

EGWordは日本語入力システムとしてEG

Convertという独自のシステムを用いている。この操作方法がすごい。ASK68Kの作法をまったく無視している。Macintosh版のユーザーが楽に移行できるようにとの配慮だろうか？　それならMacintosh版を使えばいいか。謎だ。

日本語モード/英数字モードの切り替えが［CTRL］＋スペースキー。モードは右上。文節移動はカーソルキー。文節の切り直しは［CTRL］＋カーソルキー。全角/半角変換は［XF3］＋スペース，カタカナ/ひらがな変換は［SHIFT］＋［XF1］＋Zといった割り当てだ。こうしたものは，日本語ワープロのものすごく基本的な操作だけに，ここまで違うとかなりストレスがたまる。最悪なのは手を頻繁にホームポジションから離さなくてはならないこと。

キーバインドには，ATOKやVJEなどいくつかの種類が用意されているのだが，ユーザーの指定による変更はできない。

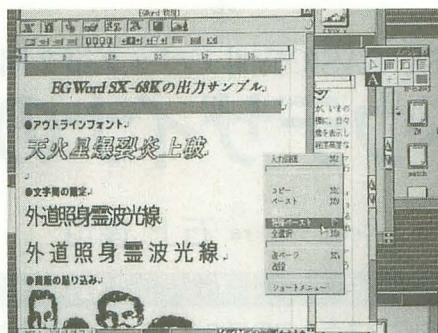
慣れればすむことという意見もあろう。それは一応正しい。正しいが，私はこの作法に慣れたいとは思わない。それでもこのレビューを書いていくうちに少しずつ慣れてしまったが，文節の切り直しはいまだに慣れない。指がASK68Kの作法に染まっていることを再確認した次第。

ただ，ED系のキーバインドに慣れていれば，キーボードを見ずに文章が入力できるようになる可能性はある。いわゆるダイヤモンドカーソルがカーソルキーに対応しているので，それを駆使すればどうにか，かな漢字変換もホームポジションで行えるだろう。

なお，EGConvertはほかのソフトでは使えない。漢字Talk7と異なり，SX-WINDOWはシステムレベルでインライン変換をサポートしていない。そのことを考えると，EGConvertがEGWordに内蔵されているという実装もやむをえないのかもしれないという気もする。

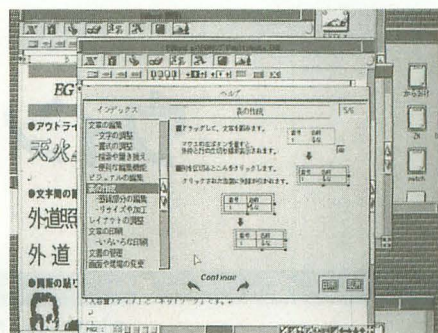
ASK68Kとはまったく違う日本語変換システムなのだから，ユーザーインタフェイス部分も独自なのはわかるが，この操作体系がASK68Kよりも操作しやすいと考えているならそれはそれで問題であろう。

もしもEGConvertのキー操作が我慢できないならば，かな漢字変換のみをASK68Kで行うことも可能である，ただし，上の理由からインライン変換は使えない。

そのほか細かいことだが，ローマ字入力モードでの拗音の入力が不安定だ。マニュアルでは“X”に続けて母音を入力するようになっているのだが，たとえば，“ぃ”の場合，“XI”では“んい”のようになってしまうことがある。“X”1文字で“ん”が入力できるほうがよいのだが。なお，拗音入力にシフトキー併用というのは対応してないようだ。

アウトラインフォントの例

ヘルプ機能は充実している

まとめ

EGWordは，日本語ワープロとしての表現力はある。しかし，入力に対するレスポンスがいいとはいえない。加えてSX-WINDOWアプリケーションとしてはやや特殊なユーザーインタフェイス。

はっきりいってX68000/030とASK68Kを手の延長のように使いこなしているような人にはすすめられない。ベタ書きの文章をひたすら打ち込むというタイプの使い方は難しい。もの書きにはちとつらいといえよう。綺麗な文書を作成して印刷するという用途にはあっている。

少々厳しいことを書いてしまったが，今回評価したのが製品バージョンであることを考えれば見過ごせない部分が少なからずある。値段も決して安くはないのだ。

図1　出力サンプル

21 世紀を先取り

新しくやってくる21世紀では、ますます情報化が進み、「組織と組織」、「人と組織」、「人と人」とのネットワークが重要になります。そんな状況のなかでは、"Communication"が効果的に演出されなければ情報の伝達はスムーズに行われません。

確実に意志や思いを伝達するためには、相手にわかりやすい書類を作成することはもちろんですが、旧来からの書類というかたちにとらわれない、新しい形態のプレゼンテーションメディアが必要となります。いわゆるマルチメディア機器が注目されているのも、そういった役割を人々が期待してのところだと思われます。

世間では盛んにマルチメディア、マルチメディアという言葉が使われていますが、過去の「ニューメディア」、また、新しいところで「バーチャルリアリティ」などのご多分に漏れず、言葉ばかりが先行して実体が知られていないのが実情です。

マルチメディアという言葉を字句どおりに解釈すると「複数の」「媒体(情報媒体)」という意味になります。複数のメディアを統合し、新しいコミュニケーションの方法を産み出そうというムーブメントの合言葉、それが「マルチメディア」なのです。

「複数のメディア」とひと言でいってしまうといかにも漠然としていますが、いまのところ、業界では「映像」と「音声」の2つのメディアを統合することを目標に、日々ディベロッパーたちがしのぎを削っています。実は、パソコンにとって、映像を表示したり音声を出力したりすることはそれほど難しいことではありません。ある程度高度なハードウェアを持ったパソコンならば、そういったことはずいぶん前から可能ではあったのです。新型パソコンの発表会では、どこでもそういったデモンストレーションを行っていたものでした。

しかし、マルチメディアは単なるデモンストレーションとは違います。マルチメディアは、新しいコミュニケーションを創造するという壮大な、そしてラディカルなビジョンなのですから。それを利用することによって、人と人がより理解しあえるようになること、この点を押さえることができなければ、マルチメディアを自称することは許されないといってもいいでしょう。

現在にいたってマルチメディアが大きくクローズアップされてきたのは、マルチメディアと呼ぶに相応しい役割を担えるだけの力を、パソコンと、そして社会が備えるようになった、あるいはその準備ができたという事情によります。その「力」の2本柱が

SX-BASIC公開デバッグ第2回

ウィンドウデザイナの使い方

Ishigami Tatsuya 石上 達也

S X - W I N D O W

今回はSX-BASICシステムでのプログラミングの基本となるウィンドウデザイナを使ってみましょう。また，レファレンスマニュアルに掲載されていなかった情報についても触れておきます。

3月号のおわびと訂正

3月号に収録されたSX-BASICに以下のようなバグが見つかりました。

●iocs関数，A_line関数が暴走する。また，プログラムの箇所によっては,「未登録の変数です」というエラーが表示される。

●SX-BASICがアイコン状態でも，メニューに「To Icon」と表示されてしまう。

●文字列を3つ以上連結しようとすると，暴走する。

●float型定数が正しく扱えない。

デバッグを行うための差分データは，この原稿を書いている時点で23Kバイトほどあり，とても掲載できる量ではありません。さらにデバッグを行ったバージョンが来月号の付録ディスクに掲載される予定ですが，現在の最新バージョンへの差分データを，MIYA-Netなどに登録させていただきましたので，これらのネットワークに参加されている方はご利用ください。

また，4番目に挙げたものに関しては，パソコン通信をしていなくても，SX-BASICを再コンパイルできる環境があれば，ファイル「SXBASIC.H」内のGVAL構造体定義中の，

```
    float              valf;
```

というのを，

```
    double             valf;
```

に置き換えることにより，デバッグできます。

マニュアル補足

また，3月号の原稿入稿後（つまりマニュアルの締め切り後)，いくつか拡張された命令，関数などがあるので，あわせて説明します。

●システム変数

・taskid

SX-BASICのタスクIDを返します。

●関数

・varhdl(int型1次元配列変数名)

戻り値：ハンドル (int型)

メモリマンからメモリハンドルを取得し，引数として指定されたint型1次元配列変数の要素を転送します。関数の戻り値として，そのハンドルの値を返します。

●プロパティ

・height

このプロパティはテキストとレクタングルが持っていますが，そのほかにwindowにも持たせました。

ウィンドウエンジンのウィンドウの縦の長さを設定します。

例） window.height=200

・width

ウィンドウエンジンのウィンドウの横幅を設定します。

このプロパティを持つアイテムはwindowのみです。

例） window.width=300

●メソッド

・put　i

　i　INT型

ビットマップアイテムの表示するデータの収められているハンドルを指定することにより，表示データの変更を行います。再描画はウィンドウエンジンが行いますので，プログラムで明示的に行う必要はありません。

このメソッドが有効なアイテムは，ビットマップのみです。

例） Bitmap1.put = varhdl(s)

　sは，int型1次元配列

また，46ページ中段「drag」の項目での例は正しくは以下のとおりです。

```
例） window.drag = 1
     プログラム
     func File_Drop(filename;str)
         print filename;"がドロップされました"
     endfunc
```

今月のメニュー

今月は，第2の顔であるウィンドウデザイナの説明をします。基本的には1993年10月号の「秋祭りPRO-68K」に収録されたものの改良版となっています。以下の記事も基本的には，93年10月号の文章に加筆・訂正を行ったものとなっているので，一部重複している箇所があります。あらかじめご了承ください。なお，プログラム作成に関して，前回に引き続きシャープ製「SX-WINDOW開発キットβ版」に収録されていた「サンプルドロー」を大幅に参考にさせていただきました。制作者の方々に厚くお礼申しあげます。

ウィンドウデザイン

コマンドシェル上では，文字は基本的に左から右へと一方的に流れ，適当なところで改行し，また一方的に左から右へと流れていきます。カーソルの位置を明示的にいじってやらないかぎり，この規則は守られます。

それに対し，SX-WINDOWのウィンドウでは，あっちこっちにいろいろな文字が書かれていたり，ボタンがあったりします。文字の大きさも一定ではありませんし，さまざまな種類や大きさのボタンが縦に並んでいたり，横に並んでいたりします。

SX-WINDOWにも，カーソル位置というか，ペン位置のような概念はあるのですが，こだわりすぎると面白味のないウィンドウになってしまいます。文字列を表示する場合には，たいてい，一緒にその座標も設定してやります（X-BASICでいうと，PRINT文はLOCATE文とペアになって出てくる，ということです)。

BASICでプログラムする場合は，画面のデザインといっても，文字の大きさは同じですし，16ドット単位でしか指定しませんし，わりと変更が簡単ですし，というわけで，実行しながら適当に改造していけばいつのまにかプログラムはできあがっていました。

しかし，SX-WINDOWの画面はそんな具合にはデザインできません。

まず，扱う情報の量が違います。同じ文字列を表示するにしても，背景色，描画色，フォントの種類……と，決めなければいけないことが，どばっと増えます。

では，どのようにすれば画面のデザインが楽に行えるようになるかといえば，話は簡単で，それこそWYSIWYG (What You See Is What You Get＝見たものそのものが手に入る) すればいいのです。

適当にウィンドウをデザインしてみて，「あ，この文字列はもう少し左だったほうがよかったな」

と思えば，マウスで文字列をひょいっとつまみ，少し左にずらしてやることができれば，かなりウィンドウのデザインは楽になるはずです。

ウィンドウデザイナの位置づけ

3月号で説明したように，SX-BASICとウィンドウエンジンがあればSX-WINDOWのプログラムは作成できます。ウィンドウエンジンにアイテムを配置するには，newメソッドを使用します。

しかし，これらのことをプログラミングするためには，アイテムの配置に関する座標データをすべてXY座標の数値データで管理する必要があります。これでは，せっかくのSX-WINDOWのGUI環境が台なしです。

そこで，アイテムの配置をGUI環境で視角的に扱うために，ウィンドウデザイナを使用します。

アイテムの配置，変更，削除

まず，WIND.Xを起動します。

すると，「Untitled」と書かれたまっさらなウィンドウと，「ツールボックス」と書かれた少し小さなウィンドウが表示されているはずです。

大きなウィンドウが，これから作業を行うカンバス（みたいなもの）で，小さなウィンドウがその際に使う道具を収めたツールボックス（みたいなもの）です。ツールボックスの上から適当な画材を選び，カンバスの上にパタペタと張っていきます。

ツールボックスはカンバスのサブウィンドウになっていて，カンバスのウィンドウがアクティブになったときのみ，画面に表示されます。

作業を行うには，まず，カンバスの右下端をマウスでつかみ，適当な大きさになるようにドラッグしてください（この作業は，一連の作業の中でいつでも行えますが，いちおう縁起物ということで）。

ツールボックスの中から，適当なアイテム（ここでは，矢印とDEL以外）をマウスの左ボタンで選びます。

そして，そのアイテムをカンバスのどの部分に配置するかを指定します。カンバスの一部をマウスの左ボタンでクリックすることによって座標を決定し，そのままドラッグして大きさを決めてやります。

もし，そのようにして配置したアイテムの大きさや配置が気にいらなかった場合には，ツールボックスから矢印のアイテムを選択し，変更を加えたいアイテムを左クリックします。すると，選ばれたアイテムは，その周囲を8個の小さな正方形とともに黒い枠で囲まれます。これがアイテムが選択された（アクティブになった）というサインです。

選択したアイテムを移動したい場合には，その枠の中心をマウスの左ボタンでドラッグします。ドラッグしようとすると，ウィンドウ上からアイテムが消え，代わりに消えたアイテムと同じ大きさを持ったグレーの枠がマウスの動きにあわせて移動するので適当な場所までドラッグしてください。

マウスの左ボタンを放せば，グレーの枠は消え，先ほどのアイテムが現れます（デスクトップ上でのファイルアイコンの扱いと同じです）。

また，アイテムの大きさを変更する場合には，そのアイテムの周囲に表示されている小さな黒い正方形（以後，ハンドルという）をマウスでドラッグしてください。アイテムの移動と同じ要領で，アイテムの大きさが変更されるはずです。

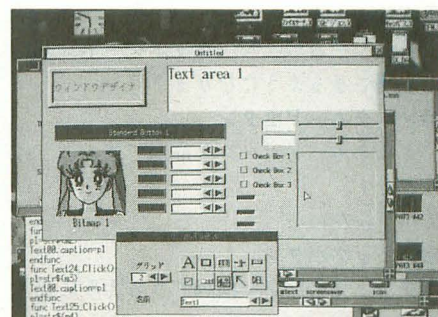
アイテムを削除したい場合には，移動，大きさ変更と同じように削除したいアイテムを選択し，ツールボックス中のDELボタンを押してください。

アイテムを自由に配置

図1 ツールボックス

ツールボックス

ツールボックスに収められた道具の種類には次のようなものがあります。これは，このウィンドウデザイナで扱える，アイテムの種類です（左上から順に右へ）。

●テキスト

文字列を張り込む枠です。文字列そのものではありません。枠を配置したあとで，プロパティ設定（後述）を行い，枠の中に文字列を注ぎ込んでやる，というスタイルを取ります。

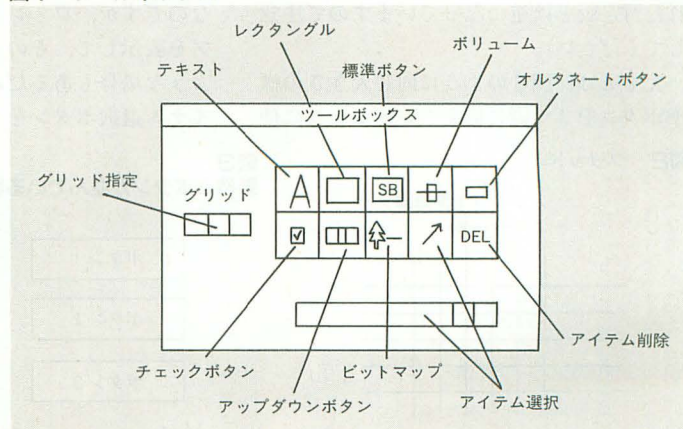
枠の中には，文字列を1行だけ入れてやることができます。

●レクタングル

長方形の枠を描画することができるようになります。複数のスイッチをひとまとめにしたり，ウィンドウのデザインにメリハリをつけるときに使用するとよいと思います。

また，このウィンドウデザイナでサポートされていない機能を使おうとする場合に，とりあえず座標を目立てておくのに便利です。

●標準ボタン

標準ボタンを描画できるようになります。

●ボリューム

ボリュームボタン（スライダーボタン）を描画できるようになります。

●オルタネートボタン

オルタネートボタンを描画できるようになります。

（下段左にいって）

●チェックボタン

チェックボタンを描画できるようになります。

●アップダウンボタン（数値調整ボタン）

アップダウンボタンを描画できるようになります。

●ビットマップ

絵を表示できるようになります。ここで表示できる絵とは，SX-WINDOW上のパターンエディタ，またはEasypaintSX-68Kで作成したPAT4形式のプロットデータです。

絵はすぐには表示されませんが，後述するプロパティ設定を行うことによって，表示されるようになります。

●アイテム選択

すでに配置されたアイテムを移動したり，サイズを変更したり，プロパティを設定/変更するときに使います。このツールを選択している状態で，マウスカーソルを特定のアイテムの上にのせて，左ボタンを押すとそのアイテムが選択されます。

●アイテム削除

アイテムが選択されているときに，このボタンを押すとそのアイテムを削除することができます。

●グリッド

このグリッドと書かれたアップダウンボタンで，ウィンドウデザイナのグリッド間距離（単位はドット）を設定します。グリッド間距離とは図2に示すような2つの直線の距離です。この設定はキーボードのSHIFTキーを押すことにより無効化されます（前回の「秋祭りPRO-68K」に掲載された暫定版とは逆になっていますので注意してください）。

たとえば，図3のように同じ大きさの標準ボタンを3つ縦に揃えて並べるときに使います。普通にマウスを操作すると，このように配置することは困難です。不可能ではありませんが，ディスプレイ上の1ドットの差を読み取れる眼力が必要となってきます。1ドットの差が読み取れない人でも10ドットの差ならなんとかわかるでしょう。で，このグリッド指定を10に設定します。

そして，通常と同じようにアイテムを配置します。ウィンドウデザイナはマウスの座標を10の整数倍としてしか認識しないようになります。つまり，アイテムは，

(10×N, 10×M)

のような座標にしか配置できなくなるわけです。

●アイテム選択ボタン（アップダウン）

通常，アイテムの選択は，選択ボタンを押して，マウスの左ボタンを押すことによって行います。ところが，このような方法では選択できないアイテムが存在する場合があります。

実は，このウィンドウデザイナでは，内部処理でアイテムが「奥行き」を持っています。奥行きといっても，3次元的なものというよりは，SX-WINDOWでいうところの「ウィンドウ優先順位」くらいに考えてください。SX-WINDOWでは，グラフィックウィンドウのような大きなウィンドウの下に隠れてしまった小さなウィンドウを直接マウスでいじることはできません。手前のウィンドウのほうが優先順位が高いからです。

このようなときには，手前のウィンドウをいったんどこかにずらしてやるか，ページアイコンをいじってやればいいのです。

これと同じように，図4のように大きなアイテムの下に隠れてしまった小さなアイテムを選択するには，一度大きなアイテムをどこかにずらしてやるか，このアイテム選択ボタンのボタンをいじってやるかの2通りの方法があります。

アイテムは基本的に重ねあわせないものなのですが，ウィンドウ全体にビットマップを表示して，その上にボタンを配置するような場合もあるだろうと思って，このアイテム選択ボタンをつけておきました。

なお，アイテムを作成すると自動的に名前がつけられますが，その名前はこのアイテム選択ボタンに表示されているとおりです。

ここでの名前のつけ方は，テキストフィールドだったら，Text1，Text2……というふうになっています。もし，この名前が気にいらない場合は，この名前が表示されている部分が編集可能な文字列となっていますので，マウスの左ボタンを押しキャレットが表示されたら，キーボードから適当な名前を入力してください（ただし，この作業は必ず後述のコード入力よりも先に行ってください）。

プロパティの設定

以上で，アイテムを画面に配置することができるようになりました。ここで，切り上げてもいいのですが，もう少し，実際のウィンドウに近い環境でデザインを行えるように，アイテムのプロパティを設定できるようになっています。

たとえば，文字列の色を変更してみたり，標準ボタンの中に「確認」だとか「取消」のような文字を入れると，かなり雰囲気を出すことができます。

このように，文字列（以下キャプション）や色のなどアイテムに付随する情報をプロパティと呼びます。設定できるプロパティの種類はアイテムによります。

ウィンドウデザイナからプロパティを設定するには，以下のように行います。

1) まず，プロパティを設定したいアイテムを選択する。

2) そのアイテムが黒い枠で囲まれ，周囲にハンドルが表示されたら，マウスの右ボタンでメニューを表示させる。

3) このメニューから，「Propaties」という項目を選ぶ。

すると，ツールボックスのウィンドウが画面から消えて，代わりにプロパティ設定用のウィンドウが画面に表示されます。ここには，設定できるプロパティの一覧が表示されているのですが，その数・種類はア

図2 グリッド

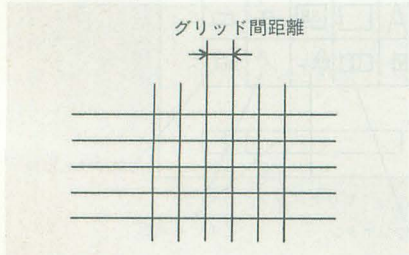



図3
整然とボタンが並んでいる場合

ボタン1

ボタン2

ボタン3

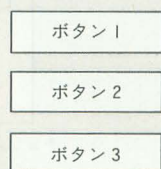
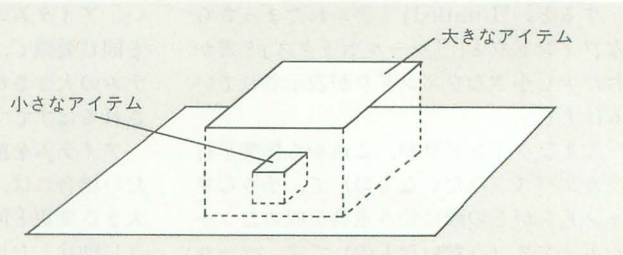



図4 マウスで選択できないアイテム

イテムにより異なり，以下のようになっています。

●テキスト

文字列，フォントの種類，行揃えモード（左寄せ，中央詰め，右寄せ），背景色，描画色，編集可フラグ（キーボードによる編集を許すかどうか），テキスト枠描画フラグ（枠を描画するかどうか）が設定できます。

●レクタングル

「彫り」の深さが－5～5の範囲で設定できます。この値が正の場合には，手前に飛び出しているような長方形に，負の場合には，奥に引っ込んでいるような感じの長方形になります。

●標準ボタン

キャプションが設定できます。

●ボリューム

最小値，最大値，初期値を設定できます。このとき，

最小値≦初期値≦最大値

という関係を満たしていなければなりません。

●オルタネートボタン

初期状態（初期値）が設定できます。

●チェックボタン

キャプション，初期状態が設定できます。

●アップダウンボタン

キャプション，最小値，最大値，初期値，編集可フラグが設定できます。

このとき，

最小値≦初期値≦最大値

を満たしていなければなりません。

●ビットマップ

指定された座標に表示するPAT4形式の絵のデータが入ったファイルを指定することができます。ファイルを指定すると，アイテムの大きさは，強制的に絵の大きさにあわせられます。以後，大きさの変更はできなくなります。

●ウィンドウ

今回のバージョンからは，ウィンドウ自身のプロパティも設定できるようになっています（アイテム名は「window」）。

なにもアイテムを選択していない状態で，カンバスのタイトル部分へマウスを持っていくと，右ボタンでファイル関係の項目と「window」のプロパティ設定の項目を持ったメニューが開きます。ここで，上から3番目の「Property」という項目を選択すると，「window」に関するプロパティを設定することができます。

現在のところ，「window」は，グローボックスフラグ（ウィンドウの大きさを可変にするか），ドラッグ可能フラグ（ファイルアイコンのドロップを受け入れるか）の2つのプロパティを設定できるようになっています。

関数（プログラム）の記述

以上の操作により，アイテムに対し，「なにを」「どこに」「どのように」という情報を与えることができました。

SX-BASICでは，さらに「なにをするために」という情報を与えることができます。

具体的には，イベント（囲み参照）が生じた場合に呼び出されるX-BASICの関数を入力します。

プロパティの設定と同様に

1)　まず，プログラムを入力したいアイテムを選択する。

2)　そのアイテムが黒い枠で囲まれ，周囲にハンドルが表示されたら，マウスの右ボタンでメニューを表示させる。

3)　このメニューから，「Code」という項目を選ぶ。

すると，ツールボックスのウィンドウが画面から消えて，代わりにプログラム入力用のウィンドウが画面に表示されます。このウィンドウは，ごく普通のテキストエディタになっています（以下，プログラム入力エディタ）。

テキスト中には，関数の初め（func ～～）と終わり（endfunc）はすでに入力されており，プログラマは，この間に必要な命令・関数を挿入するだけで，関数が完成するようになっています。

ここで入力する関数の名前は基本的に，

アイテムの名前＋_＋イベント名

となっています（表1参照）。

プロパティ設定用のウィンドウと違って，プログラム入力エディタは，デスクトップ画面上に複数開いておくことも可能となっています。つまり，「Text1」のコードを見ながら「Text2」のコードを入力する，といったことができます。もちろん，これらのウィンドウ間でカット＆ペーストを行うことも可能です。

また，ウィンドウデザイナには，プログラム入力エディタの内容が適宜フィードバ

表1　アイテムごとに必要な関数

●テキスト

「アイテム名」_KeyIn(s;str)
　(editプロパティ＝1の場合のみ)

アイテム中でカーソルが点滅しているときに，リターンキーが押されると呼び出されます。

引数には入力された文字列が代入されます。

「アイテム名」_Click()
　(editプロパティ=0の場合のみ)

アイテムをマウスが指している場合に左クリックされると呼び出されます。

引数はありません。

●レクタングル

「アイテム名」_Click()

アイテムをマウスが指している場合に左クリックされると呼び出されます。

引数はありません。

●標準ボタン

「アイテム名」_Click()

アイテムをマウスが指している場合に左クリックされると呼び出されます。

引数はありません。

●ボリューム

「アイテム名」_Change(i;int)

マウスから値に変更が加えられると呼び出されます。

引数に変更された値が代入されます。

●オルタネートボタン

「アイテム名」_Change(i;int)

マウスから値に変更が加えられると呼び出されます。

引数に変更された値が代入されます。

●チェックボタン

「アイテム名」_Change(i;int)

マウスから値に変更が加えられると呼び出されます。

引数に変更された値が代入されます。

●アップダウンボタン

「アイテム名」_Change(i;int)

マウスから値に変更が加えられると呼び出されます。

引数に変更された値が代入されます。

「アイテム名」_KeyIn(s;str)
　(editプロパティ＝1の場合のみ)

アイテム中でカーソルが点滅しているときに，リターンキーが押されると呼び出されます。

引数には，入力された文字列が代入されます。

●ビットマップ

「アイテム名」_Click()

アイテムをマウスが指している場合に左クリックされると呼び出されます。

引数はありません。

「アイテム名」_Menuselect(i;int)
　(menuプロパティ≠""の場合のみ)

アイテム上で，右ボタンが押されることによりメニューが表示されます。このとき，メニュー中のいずれかの項目が選択されると，この関数が呼び出されます。

引数に，選択された項目の除数が代入されます。

●ウィンドウ

window_Menuselect(i;int)
　(menuプロパティ≠""の場合のみ)

ウィンドウエンジンのタイトルバー上で，右ボタンが押されることによりメニューが表示されます。このとき，メニュー中のいずれかの項目が選択されるとこの関数が呼び出されます。

引数に選択された項目の除数が代入されます。

File_Drop(s;str)
　(window.drop＝1のときのみ)

ウィンドウエンジンに対して，ファイルアイコンがドロップされたときに呼び出されます。

引数にファイル名が代入されます。複数選択されたときには，アイコンリストの初めに登録されたファイル名が代入されます。

ックされているので，いちいち「設定」ボタンを押して……などというような動作は必要ありません。ウィンドウデザイナは，必要なとき(たとえば，ファイルのセーブ)には，つねに最新の情報を取得しています。

「設定」ボタンがない，ということは「取消」ボタンもない，ということですので，大量の「カット」を行うような場合には，シャープペンなどに，その内容をペーストして保存してください。

また，ウィンドウデザイナは，各プログラム入力エディタからテキストを取得していますが，その内容にはまったく関与しません。よって，ここではまったく関係のない関数を記述することもできますし，極端な話，C言語プログラムを書き込んでもかまいません（SX-BASICには渡せなくなりますが)。

同じ理由で，ウィンドウデザイナにより作成された関数の名前を変えるときには，必ずツールボックス上からアイテムの名前も同じように変更してください。そうしないとアイテム名に関してSX-BASICとウィンドウエンジンでの整合性が保てなくなります（仮引数の名前は，自由に変えてかまいません)。逆に，ツールボックス上からアイテムの名前を変えたときには，関数の名前も同じように書き換えてください。

関数(プログラム)の順序

このように，ウィンドウデザイナでは，プログラム入力エディタによって，アイテムごとに対応する関数を入力します。たくさんの関数を入力していけば，やがてそれらは，ひとつのプログラムになっていくのですが，その前にひとつ問題があります。

プログラムには，たいてい初期化処理と呼ばれる，どの関数よりも初めに実行されなければならない処理があります。また，SX-BASICでは，グローバル変数の宣言は，さらにその前のすべての命令に先立って行われなければなりません。

X-BASICでは，関数の名前とは関係なく，初めに書かれたところからプログラムは実行されていきます。

しかし，ウィンドウデザイナで扱うデータは，基本的にアイテムが平面的に並べられただけですから，初めや終わりといったことにたいした意味はありません。

ウィンドウデザイナでは，ウィンドウのプログラム入力エディタで入力されたテキストがSX-BASICに引き渡された際，最初に読み込まれるようになっています。

SX-BASICでは，グローバル変数の宣言はどんな命令よりも先に行われなければなりません。つまり，ウィンドウデザイナですべてのプログラムを管理する場合，変数宣言や初期化処理は，この「window」のコード入力エディタで行う必要があります。

3月号43ページのリスト1を見てください。

「▼」マークで始まる行がアイテムの情報を示し，それに続くプログラムがウィンドウデザイナ上から入力されたもので，次の「▼」マークまで続く，といった構造になっています。

このように，ウィンドウデザイナはどのようにアイテムを配置しても，ウィンドウのデータやプログラムをいちばん初めに出力するようになっています。

プログラム中に，「ここで初期化に必要な処理を行ってください」と書かれたコメント文がありますが，まさに，ここで書かれた命令がSX-BASICで最初に実行されるわけです。

ファイルの保存

なにもアイテムを選択していない状態で，カンバスのタイトル部分へマウスを持っていくと，右ボタンでファイル関係の項目と「window」のプロパティ設定の項目を持ったメニューが開きます。ウィンドウデザイナの動作に関する項目も，このメニューの中に収められています。

メニューの項目を上から説明すると，

●Load File

ファイルに保存したデータを読み込みます。この項目を選択すると，ファイルネームを聞いてくるウィンドウが開きますので，そこにロードしたいファイルの名前を入力してください。このとき，ファイルアイコンがデスクトップ上にあれば，ドロップしてもけっこうです。

●Save File

現在のアイテム，そのプロパティの内容とプログラム，ウィンドウの大きさなどの情報をファイルに出力します。

操作はLoadと同じです。

●Property

「window」のプロパティを設定します。

●Code

エディタウィンドウが開きますので，ここにSX-BASICのコードを記述できます。

SX-BASICでは，ここに記述されたコードが最初に実行されますので，グローバル変数の宣言，初期化処理を行ってください。

●終了

ウィンドウデザイナを終了します。ファイルのセーブを行っていない場合には，確認のダイアログが開きます。

イベントとイベントドリブン

ウィンドウ上に標準ボタンが置かれている場合を考えてください。このボタンの目的は，マウスのクリックを通じて，操作している人の考えをプログラムに伝えることです。

X-BASICなら，ループ中でマウスの座標を監視するようなプログラミングを行います。マウスのクリックを検出するまでは，まったく無駄なコードを実行していますが，基本的にシングルタスク環境ですから，それによってほかのプログラムに迷惑をかけることはありません。

ひるがえって，SX-WINDOWのようなマルチタスク環境では，同時に複数のプログラムが実行されている可能性があります。限られた処理能力を，実行されているプログラムの頭数で割るわけですから，あるプログラムでまったく無駄なことをやられては，ほかのプログラムに迷惑がかかってしまいます。

そこで，SX-WINDOWでは，「マウスやキーボードが押されるのを待つ」という行為をシステムで一括して行っています。

プログラムはループ中でマウスの監視を行う代わりに，システムに対し，「マウスが押されたら呼んでねー」とひと声かけて，処理を中断します。システムは，マウスのクリックを検出したら，先ほどのプログラムに対し，「マウスが押されたよー」と情報を渡してやります。この情報を受け取ったプログラムは実行を再開し，必要な処理を行います。

この際，システムが発行する「マウスが押されたよー」という情報を「イベント」といいます。システムは，マウスが押された，キーボードが押された，というような出来事(イベント)しか報告しないからです。

つまり，SX-WINDOW上のプログラミングは，初期化処理を除くと，

マウスがクリックされた → 処理1
キーボードが押された → 処理2
アイコンがドロップされた→ 処理3
⋮

という巨大な対応表を作り上げることにほかなりません。

システムがイベントを与えて，アプリケーションがそれを処理する，逆にいうと，アプリケーションを動かすには，なにかイベントを与えてやらなければならない，ということで，このようなシステムをイベント駆動型（イベントドリブン）のシステムといいます（ここで，「嘘つけー。『暁子．X』は，キーボードもマウスも押してなくても動き続けるじゃないか」とあなたはいうかもしれない。しかし，あれだって「なにも起きてないよ」というイベントをシステムが送っているので，なにも起きてなくても，そのイベントに対応して，自転車と犬は前に進むのです)。

varhdl関数とputメソッドについて

varhdl関数とputメソッドはSX-BASICのなかでもひときわ特殊な機能です。

現在，SX-BASICにはウィンドウ中に直線や円を描画する機能は持ちあわせていません。将来拡張する予定はありますが，ある程度の拡張では，カバーしきれない範囲がどうしても残ってしまいます。

ちなみに，どういう点で思案中なのかというと(多少専門的になってしまいますが)，ウィンドウのアップデートに関してです。SX-WINDOWのアプリケーションは，ご存じのように，いつでも自分のウィンドウ内に描かれた文字や図形を再現できるよう情報を蓄えておく必要があります。もちろん，このようにX-BASICにはないSX-WINDOWだけの都合によるものは，ウィンドウエンジンで辻褄をあわせるようにするべきです。

一般には，このような場合，スクリプト（手順書）と呼ばれる手法が用いられます。

たとえば，

座標(10,10)から(30,30)に線分を引け
座標(15,15)を頂点として半径10の円を描け
座標(12,34)に文字列「こんにちは」と描画しろ
⋮

といったかたちのものです。

ウィンドウ内に描画する際には，このような命令書をどこかに控えておきます。で，必要になったときには，この命令書を左手で見ながら，右手で描画を行うのです。ウィンドウ内に単純な装飾を行うだけでしたら，この方法にまったく問題はありません。

次にような機能を考えてください。

ゲームのように，ディスプレイ上のキャラクターを移動させようとした場合，一般的には，描いてあったキャラクターを一度消去し，少しずらした位置に描画します。

これをスクリプトに記録したとしましょう。

(10,10)にキャラクターを描画する
(10,10)のキャラクターを消去する
(11,10)にキャラクターを描画する
(11,10)のキャラクターを消去する
(12,10)にキャラクターを描画する
⋮

というデータが延々と続いてしまうのです。

さて，もうひとつの方法として，ビットマップデータを丸ごと記憶しておく，という手法があります。スクリプトによる記憶方法を画像のドローデータによるものとすると，この方法はペイントデータによるものということができます。

ウィンドウの状態を保存しておくために十分な大きさのメモリ領域を確保しておき，描画を行う際には，実際の描画を行うとともに，必ずそのメモリ領域にも描画を行います。

(10,10)という座標に点を打ったら，そのメモリ領域の対応する点のデータもいじってやります。

この方法だと，キャラクターを移動させようが，いくら消去と描画を繰り返そうが，使用するメモリ容量は一定です。

さて，次のようなプログラムを考えてください。

```
for x=0 to 10
    for y=0 to 10
        pset(x * 2, y * 2, 1)
    next
next
```

線や円弧で表すことのできない図形は，このように点の集まりとして表現します。

中心と半径を与えて描いた円と，点を100回指定して描く図形と，使用する画素の数では同じかもしれませんが，前者は実画面から控えの画面へのデータ転送を1回行えばよいのに対して，後者では100回も行わなくてはなりません。

ここでいうデータ転送というのは，基本的に描画領域全体が範囲となっていますから，その量はかなりのものです。転送する量が大きければ大きいほど実行時間がかかりますから，ウィンドウ内には，うかつに図形を描画することもできなくなってしまいます。

このように，絵の保存には2種類のやり方があって，アプリケーションによって方法を選択しなければならないのですが，ウィンドウエンジンのようになにに使われるかわからないシステムアプリケーションでは，方法を固定してしまうと，かえって用途が限られてしまうおそれがあります。

このようなジレンマの解消策として，デバッグ担当の（？）中野氏によって締め切り直後に考案されたのが，putメソッドによる表示法です。

ここでいうputメソッドとは，X-BASICのput命令と同じような機能です。引数の値をメモリハンドルと見なし，そこに収められている内容をビットマップデータ（正確にはプロットデータですが，詳細はまたの機会に）とみなし画面に表示する，という機能です。

X-BASICには，メモリハンドルで示されるデータを扱うには便利な命令がありませんので，putメソッドとあわせて，拡張された組み込み関数がvarhdlです。

この関数は，メモリマンからメモリハンドルを取得し，そこに引数として与えられたint型1次元配列変数の内容をコピーします。

リスト1を見てください。これは，putメソッドとvarhdl関数を使った簡単なサンプルプログラムです（ひなまつりPRO-68Kに収録）。

まず，4行目から配列変数aにビットマップデータを代入していきます。手元に適当な大きさのデータがなかったので，ここではウィンドウデザイナのツールボックスに用いた「A」の文字を使用しました。

38行目からのfor〜nextループでそのデータを少しずつ0で埋めていきます（最初の2つのデータはビットマップデータの大きさを表しているので，これをいじるとその後に続くデータが破綻してしまう）。

連続写真を1コマずつ掲載するわけにもいきませんので，とりあえずリスト2を実行してください。

このプログラムでは，データを1カ所書き換えるごとに，再描画を行っていました（40行目にあるループ中のputメソッドがそれ）。

リスト2では，ループの外にputメソッドを置きました。実行するまでもないと思いますが，一瞬にして「A」の文字が消えます。

以上のように，図形の描画を行う際には，putメソッドを用いて明示的にウィンドウ内への描画のタイミングを作成してください。

次期バージョンでは，lineやらcircleやらの豊富なメソッドを用意する予定ですが（予定は未定にして……），putメソッドによる方法も残しておくつもりです。

最後に，上級者向けにメモリハンドルの生成と廃棄のタイミングについて少々。

リスト1のように，ループ内でvarhdlを用いれば，メモリエリアは消費され続けます。どこかで，これと同じスピードでメモリハンドルを開放し続けなければなりません。

現在，メモリハンドルを開放するには，A_lineコールを使って，

```
func MMHdlDispose(hdl;int)
    A_line(&hA038,hdl)
endfunc
```

のような関数を作らなくてはいけませんでしたが，リスト1ではそのような命令はどこにも見当たりません。

実は，putメソッドが伝えられると，ウィンドウデザイナは，いままでのビットマップデータを収納していたメモリハンドルを放棄して，新しく指定されたメモリハンドルに切り替わるようになっています。

つまり，リスト1において，作成されたメモリハンドルは，SX-BASICではなく，ウィンドウエンジンが破棄を行っていたのです。

鋭い方は気がついたかもしれませんが，varhdl関数というのは，一見汎用性のある関数のふりをしていて，その実，ほとんどputメソッド専用に拡張された関数なのです。

リスト1

```
 1: ▼Window Size (100,100),1,0,ﾋﾞｯﾄﾏｯﾌﾟ
 2: /* ここで、初期化に必要な処理を行なって下さい
 3: int i
 4: dim int a(200) = {
 5: &h00000000,&h001e001e,
 6: &h00000000,&h7ffffffc,&h7ffffffc,&h7ffffffc,
 7: &h7ffffffc,&h7ffffffc,&h7ffffffc,&h7ffffffc,
 8: &h7ffffffc,&h7ffffffc,&h7ffffffc,&h7ffffffc,
 9: &h7ffffffc,&h7ffffffc,&h7ffffffc,&h7ffffffc,
10: &h7ffffffc,&h7ffffffc,&h7ffffffc,&h7ffffffc,
11: &h7ffffffc,&h7ffffffc,&h7ffffffc,&h7ffffffc,
12: &h7ffffffc,&h7ffffffc,&h7ffffffc,&h7ffffffc,
13: &h7ffffffc,&hfffffffc,&h00000000,&h00000004,
14: &h00000004,&h00000004,&h00000004,&h00030004,
15: &h00030004,&h00058004,&h00058004,&h0008c004,
16: &h0008c004,&h00106004,&h00106004,&h00203004,
17: &h00203004,&h003ff004,&h007ff804,&h00401804,
18: &h00800c04,&h00800c04,&h01000604,&h03000604,
19: &h07c00f84,&h07c00f84,&h00000004,&h00000004,
20: &h00000004,&h00000004,&h00000004,&hfffffffc,
21: &h00000000,&h00000000,&h00000000,&h00000000,
22: &h00000000,&h00000000,&h00000000,&h00000000,
23: &h00000000,&h00000000,&h00000000,&h00000000,
24: &h00000000,&h00000000,&h00000000,&h00000000,
25: &h00000000,&h00000000,&h00000000,&h00000000,
26: &h00000000,&h00000000,&h00000000,&h00000000,
27: &h00000000,&h00000000,&h00000000,&h00000000,
28: &h00000000,&h00000000,
29: &hffffffff,&hffffffff,&hffffffff,&hffffffff,
30: &hffffffff,&hffffffff,&hffffffff,&hffffffff,
31: &hffffffff,&hffffffff,&hffffffff,&hffffffff,
32: &hffffffff,&hffffffff,&hffffffff,&hffffffff,
33: &hffffffff,&hffffffff,&hffffffff,&hffffffff,
34: &hffffffff,&hffffffff,&hffffffff,&hffffffff,
35: &hffffffff,&hffffffff,&hffffffff,&hffffffff,
36: &hffffffff,&hffffffff
37: }
38: for i=2 to 121
39:   a(i) = 0
40:   Bitmap1.put = varhdl(a)
41: next
42: func File_Drop(filename;str)
43: endfunc
44:
45: ▼8,Bitmap1 (36,32,66,62),0,stn.bmp
46: func Bitmap1_Click()
47: endfunc
```

リスト2

```
上はリスト1と同じ
38: for i=2 to 121
39:   a(i) = 0
40: next
41: Bitmap1.put = varhdl(a)
42: func File_Drop(filename;str)
43: endfunc
44:
45: ▼8,Bitmap1 (36,32,66,62),0,stn.bmp
46: func Bitmap1_Click()
47: endfunc
```

あなたにもできる

SX-BASICプログラミング入門

Nakano Shuichi 中野 修一

S X - W I N D O W

皆さんSX-BASICは試されたでしょうか？　使った方は意外に簡単にウィンドウプログラムが作れることがおわかりでしょう。ここではSX-BASICプログラミングの基本と今後の課題などを考えてみましょう。

デバッグ協力求む

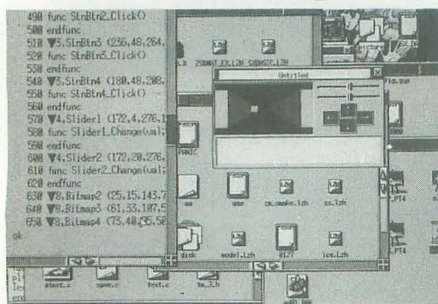
全国のSX-WINDOWユーザーの皆さん。SX-BASICはご覧になったでしょうか。これにより，従来は非常に敷居の高かったSX-WINDOW上のプログラム開発が簡単に行えるようになりました（というか，なろうとしている）。

くどいくらいに「バグが多い」と書いたためか，恐れをなした人もいるようですが，いくつかバグ報告も届きました。ご協力ありがとうございます。

そのほか「ちゃんとバグをなくしてから発表してほしい」という意見はもっともですが，バグをなくすために公開しているわけですので，そこのところをご理解して協力してくださるようにお願いします。来月号の付録ディスクでは漸進的に進歩したバージョンをお届けできると思います（まだ完成にはほど遠い）。

デバッグと拡張がちゃんぽんで進んでいますので新たなバグも入っているかもしれません。

石上氏の原稿でだいたいまとめられているとは思いますが，それ以外にテキストが20Kバイト（実質13Kバイト程度）を超えるとおかしくなる，とかファイル入出力関数がおかしいといった症状があります。

あのソースがコンパイルできる環境にある人はSXBASIC.HのTEXT_MAXのあたりの定数を増やして再コンパイルしてください。それでかなり安定した動作になるはずです。発表されてから現在までにいくつかのデバッグ版がありましたが，私はディスク収録版を定数拡大しただけのものをもっとも愛用しています。作成するプログラムの大きさが10Kバイトに満たないような場合ならディスク収録版で十分でしょう。

とりあえず画面から作る

ファイル関数については，INT以外の型でいまひとつ灰色の部分があります。ちなみに私はRAMディスク以外では書き込みを試したことはありません。

こんなに気をつけているのも，それまでに多くのバグを見つけてきているからですが（マスターアップ直前まで四則演算がちゃんと動かなかったというのに比べれば現在のものはまだ安定している），バグに慣れてくると，バグらしい症状に出合うとデバッグされるまでのあいだそれを回避するようにプログラム開発を進めていくようになるものです。

notの動作がX-BASICと違えば，

&hFFFFFFFF－n

のようにしてみたり，fseek () が動かなければ，そのたびにfopen() とfclose() を繰り返したり，変数名が検索されないときには宣言位置を前に持ってきたり，ファイル読み込みの要素数が違っていれば配列変数を大きくしたり……，なにか涙ぐましいことをやってきたような。

しかし，これではデバッグ効率はあまり上がりません。

　バグは困る

　デバッグするしかない

んで，デバッグするにはなにかプログラムを作ってみるしかないじゃあないですか。

なにを作るべきか

なにかプログラムを作ってみようと思ったとき，意外に問題になるのは「いったいなにを作るべきか」ということでしょう。

ありがちなのは，いきなりとんでもないものを作ろうとして失敗し「SX-BASICなんて使いものにならない」などと思い込んでしまうことです。これはX-BASICでもいえることですが，相手はBASICなんですから，それなりのつきあい方というものがあります。やはり最初は慣れていくところから始めなければなりません。

とかいいつつ，私が最初に手がけたのは（電卓は別として），楽譜エディタとグラフィックエディタです（ダンジョン型のRPGも検討中）。

見るからに無理そうな題材ですが，ちと面倒なだけで，実装方法によってはBASICでも記述できると判断しました。何度もいっていることですが，世の中の人が思っているほどBASICはなにもできない言語ではありません。逆説的にいえば，これくらいのものが力技でできなくては，それこそなにもできないシステムということになってしまうでしょう。

「とりあえず簡単なものから……」という消極的な立場でやっているとできあがったとしても「簡単なものだから」ということで終わってしまいます。全然面白くありません。

じゃあどうすればいいのでしょう。

なにを作るかという場合には，作りたいものを作ることがいちばんです。それにはまず，自分にとってどんなものが必要かということを知ることです。

「なにがほしい？」

と聞かれたとき，

「操作性のよいZ-MUSIC用の楽譜エディタ」

「パターンエディタよりも高機能なパターン作成用のグラフィックエディタ」

のようにすぐに答が出てくれば問題ありません。SX-WINDOWユーザーの方なら，少なからず不満点はあることでしょう。

まずは画面

いきなり作り始めてもコケますから，まずは，本当にできるのかできないのかを判断するためのテストプログラム作りから始めます。付録ディスクに入っていたなんの意味もないプログラム群が私のテストプログラム（の一部）にあたります。

この時点でできそうな気がしてきたら，途中で放り出さない限り，たいていできあがってしまうものです。この場合，行き詰まってもできないと思い込むのではなく，「できるように作る」，ということが重要です。

ここでSX-BASICが有利な点があります。それは，

まず画面を作ってしまえる

ということです。

SX-BASICのプログラミングはウィンドウデザインから始まります。機能ができていようがいまいが，完成時のウィンドウイメージを作り上げることができるのです。画面ができていれば，プログラムもできそうな気になるものです。

その意味も含めて，ウィンドウデザインはあくまで美しくあらねばなりません。

はったりでもなんでも，美しい画面というのはアプリケーションにとって非常に重要なことです。便利なソフトみたいだけどデスクトップには置いておきたくないというソフトではちょっと悲しいですから。

さらにSX-BASICで有利な点は，ユーザーインタフェイスを記述する必要がないということです。もちろん，操作に対応する各種処理は記述しなければならないのですが，そこにいたるまでの，従来いちばん面倒な部分だったものがシステムにより吸収されているのです。

SX-BASIC TEC TIPS

次にSX-BASICでプログラミングする際にポイントとなる項目をいくつか挙げておきましょう。

●プログラムの構成

▼マークのついた行がウィンドウエンジンに対する命令です。これはどこに書いてあってもプログラム実行前に処理されます。記述の順番はそのまま表示の優先順位となります。後ろに書いてあるものが優先して表示されます。

プログラムの先頭は，

▼Window Sise（……）

のように始まっています（これ以外だとウィンドウデザイナで読み込めなくなる）。

この直後に書かれたブロックはプログラム起動時に処理されます。また，ここで宣言された変数はグローバル変数になります。

それ以外のウィンドウアイテムに対する処理などはすべて関数として用意します。

●変数宣言

各関数の先頭でのみ可能です。

グローバル変数は宣言時に自動的に初期化されますが，ローカル変数は初期化されません。

●di（）とei（）

動作速度を上げるために有効なのがdi()関数とei()関数です。わかる人にはわかると思いますが，もちろん，disable interuptとenable interruptの略です。

通常SX-BASICは中間コードの１処理が終わるとシステムに戻るようになっていましたが，di()関数が実行されるとシステムに帰らずにずっと処理を続けます。

通常は，

```
for i=0 to 1000
    print i:
next
```

のような処理を行っていてもほかのSXプログラムはマルチタスクでちゃんと動いていましたが，この処理の直前にdi()を置くと，数字を表示し終わるまでほかのプログラムを止めてしまうようになります。

なお，たいていのプログラムはダイレクトモードで実行されますので，di()をしたあとには必ずei()をしなければならないというわけでもありません（関数の実行が終われば元に戻る）。

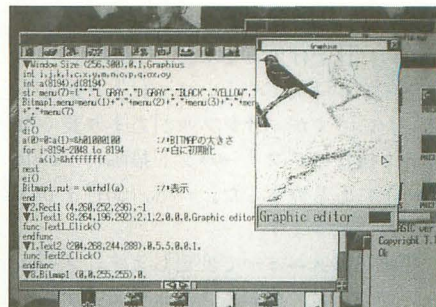
for文でまとめて配列の演算などを行う場合には前後につけておいたほうがいいでしょう。まとめて描画を行うような関数なら必ず先頭にdi() を置くべきです。SX-BASICはヌルイベントをクロックとして動作しているようなものですので，かなりの差が出てきます。

Ｃ言語で書かれているとはいえ，本来，インタプリタ自体の速度はSX-BASICのほうがX-BASICよりも高速です。

●配列への代入

X-BASICと同様，

```
int a (5)
a= {0, 1, 2, 3, 4, 5}
```

のような記述が可能です。ビットマップパターンの書き換えなどの際に重宝します。大きな配列でこの処理を行うときは事前にdi()を実行しておいてください。変数宣言部ではdi()がききませんので，プログラム中で行うほうがよいでしょう。

ちょっと無謀なグラフィックエディタ

グラフィックを扱う

それでは実際のプログラム解説に入ります。

今回プログラムを掲載するのは簡易グラフィックエディタのみです。楽譜エディタは未完成のくせに巨大なため掲載は見合わせます。ファイル入力関数の問題でまだファイルロードがちゃんと動きません。うまくいけば来月号でサンプルとして収録できるでしょう。

先月号のリファレンスマニュアルには書いてありませんでしたが，SX-BASICではビットマップファイル名を指定する方法以外でもグラフィックパターンの表示が可能です。すなわちputです。

Bitmap1.put=verhdl（A)

のようにすると，配列Aに入ったデータをビットマップとして表示します。配列Ａの中身はPAT4データそのものです（当然データは水平型で頭に８バイトのヘッダつき）。

ここではそのputを使ったサンプルプログラムを見ていきます。

putは配列の内容をビットマップとして表示するというものですが，SX-BASICで唯一，ウィンドウエンジンのグラフィック部分にアクセスすることができる機能といっていいでしょう。

理屈の上では，配列の中身を操作すれば，ラインを引くこともできますし，ペイントだって行うことができるでしょう。

大胆にいえば，

「グラフィック画面を設定できる」

といえなくもありません。ということで，グラフィック機能を使ったサンプルの最たるものといえば，やはりグラフィックエディタではないでしょうか。処理が重いということはやってみなくてもわかるのですが，どの程度重いかということのテストも兼ねて作ったものがリスト１です。

ここでは256×256ドットのスクリーンを

設定して，マウスでお絵描きができるようなものに仕上げています。

ラインルーチンくらいはつけようかとも思ったのですが，サンプルがあまり長くなるのも嫌なのでフリーハンド描画のみのサポートにとどめました。一応，50行まで入力すれば動きます。

描画色はメニューで変更します。メニューの使い方は，先頭でメニューアイテムを用意しておけば，Menuselect () 系の関数が呼び出されるようになっています。

なお，PAT4形式のファイルをウィンドウに放り込むことにより（ただし横32ドット以上のもの），画面に表示することもできます。

描画は1次元配列変数へのビット単位の書き込みとウィンドウエンジンへの表示命令に分けられます。PAT4はテキスト画面用のデータ構造ですから，プレーンごとに処理する必要があります。

リスト1

```
 1: ▼Window Size (256,300),0,1,Graphius
 2: int i,j,k,l,c,m,n,o,p,q
 3: int a(8194),d(8194)
 4: str menu(7)={"","L GRAY","D GRAY","BLACK","YELLOW","RED","G
REEN","BLUE"}
 5: Bitmap1.menu=menu(1)+","+menu(2)+","+menu(3)+","+menu(4)+",
"+menu(5)+","+menu(6)+","+menu(7)
 6: c=5
 7: di()
 8: a(0)=0:a(1)=&h01000100          :/*BITMAPの大きさ
 9: for i=8194-2048 to 8194         :/*白に初期化
10:      a(i)=&hffffffff
11: next
12: ei()
13: Bitmap1.put = varhdl(a)         :/*表示
14: end
15: ▼2,Rect1 (4,260,252,296),-1
16: ▼1,Text1 (8,264,196,292),2,1,2,0,0,0,Graphic editor
17: func Text1_Click()
18: endfunc
19: ▼1,Text2 (204,268,244,288),0,5,3,0,0,1,
20: func Text2_Click()
21: endfunc
22: ▼8,Bitmap1 (0,0,255,255),0,
23: /* ============================== ボタンが押された
24: func Bitmap1_Click()
25: di()
26: repeat
27:      pset(mousex,mousey,c)
28: until mousel=0
29: Bitmap1.put = varhdl(a)
30: endfunc
31: /*  ============================= メニュー処理
32: func Bitmap1_Menuselect(c0)
33: c=c0
34: Text2.backcolor=c
35: endfunc
36: /* ============================== 点を打つ
37: func pset(x,y,c)
38: m=x mod 32
39: o=x/32+y*8+2
40: if m>0 then p=&h40000000 shr (m-1) else p=&h80000000
41: switch c
42:      case 7  :a(o+4096)=a(o+4096)or p
43:      case 3  :a(o+2048)=a(o+2048)or p
44:      case 1  :a(o)=a(o)or p           :break
45:      case 6  :a(o+4096)=a(o+4096)or p
46:      case 2  :a(o+2048)=a(o+2048)or p   :break
47:      case 5  :a(o)=a(o)or p
48:      case 4  :a(o+4096)=a(o+4096)or p
49: endswitch
50: endfunc
51: /* ============================== file dropによるPAT4のロード
52: func File_Drop(st;str)
53: int i,j,l,fn,xs,ys,s
54: di()
55: fn=fopen(st,"r")
56: fread(d,2,fn)
57: xs=d(1)/65536:ys=d(1)mod 65536
58: l=(xs+31)/32:s=ys*l
59: fread(d,s*4,fn)
60: fclose(fn)
61: for k=0 to 2
62:      for j=0 to ys-1
63:           for i=0 to l-1
64:                a(j*8+i+k*2048+2)=d(j*l+i+s*k)
65:           next
66:      next
67: next
68: Bitmap1.put = varhdl(a)
69: endfunc
```

40行目のif文はshrのバグ回避のためのもので，本来は

p＝&H80000000 shr m

のようにすべきものです。

その後ろのswitch文がやや込み入ってますが，色コードでプレーンを描き分けるためのものです。現在は速度重視で重ね描きを考慮していませんが，このあたりはもう少しちゃんと処理する必要があります。

なお，このサンプルでは範囲に関するエラー処理をしていません。場合によっては暴走するので，ビットマップの範囲外にマウスカーソルを出さないようにしてください。

作ってみてわかったのは，「予想したほど使えないものではない」ということでした。

リアルタイムに描画していないのはウィンドウエンジンに描画範囲の指定がないからです。1ドット追加するにも全体を描き直さなければならないので，どうしても無駄が出ます。アップデートリージョンを一緒に指定してやることができればそれなりに速くなることでしょう。

いずれにしても，これだけでは絵を描くには不十分です。従来のパターンエディタには多少不満点があるので，もしかしたらこれをタイリングペイントなどを持った拡張ツールとして発展させていくかもしれません。

アイテムの動的管理

写真は開発中のSX-WINDOW用の楽譜入力ツールです。

石上氏にputは絶対必要だと力説した手前，それを有効に活用した例として作成したものです。

同時に，これは「アイテムの動的参照またはクラス管理による機能の継承は絶対必要だ」の証明にもなっているプログラムです。

SX-BASICでは，ウィンドウに表示するアイテムはnewにより新規作成することができるのですが，その「アイテムに対する操作」はすべてあらかじめプログラムで用意されていなければなりません。

SX-BASICではアイテムに対応する処理・関数はそれぞれのアイテムに対応した個別の名前で静的に管理されています。

これでは，プログラム実行中に新しい名前でアイテムを作ることはできますが，プログラム実行前にその名前を使った各種関数が一式定義されていることが必要という，一種のパラドックスが発生します。

また，各アイテムの処理内容はきわめて似通ったものになりがちです。そういった場合，ほかの関数を参照するようなことができればプログラムは非常に簡潔になります。

たとえば，クリックされると反転するBitmapが２つあったとしましょう。その場合，

```
func Bitmap1_Click ()
Bitmap1.mode=1
endfunc
func Bitmap2_Click ()
Bitmap2.mode=1
endfunc
```

のような２つの関数を記述します。

さらにこれがドラッグされたりすると，さらに複雑な，しかし似たような関数が２つ必要になります。

こういった単調なものを延々と記述していると，「ワイルドカードが使えたら……」とか，「関数を返すプロトタイプの関数があれば……」とか，「アイテム名に変数が使えたら……」といったアイテムを動的に扱う機能がほしくなります。また，オブジェクト指向でいうところの「クラス」とか「継承」「多重継承」といった概念も頭をかすめます。

個人的には関数名に変数が使用できるようになって実行時に評価してくれるならば文句はないのですが（実現もいちばん簡単そうだ），それはそれで非常に汚いシステムになりかねません。

現在の楽譜エディタは，静的な処理だとどうなるかという見本のような状態（それでもputのおかげでずいぶん簡潔になっている）です。

処理内容はといえば，当然，音符を選んでどんどん置いていくわけですが，前述のようにアイテムが最大限に増えた状況を想定してプログラムを作っておかねばなりません。

当然予想されることですがプログラムは大きくなっています。私が作るプログラムは，かなり込み入ったものでも200行を超えることは滅多にないのですが，現状では2000行へ向かってまっしぐらです。

現在，１画面中の音符25個にまで対応しています。試しに32個対応にしたときには200行くらいプログラムが長くなりました。また，現在は１トラックのみに対応していますが，ほかのトラックを参照するといった際には表示を切り換えることになるでしょう。このままで２つのトラックを許容するとプログラムが２倍の大きさになりますから。

ということで，現在，将来に禍根を残さないようなかたちでアイテムを動的に管理できるような方法を摸索中です。

SX-BASICの今後

すでに，読者からもいくつかの要望が届いています。

・グラフィックのサポート
・Ｃへのコンバータ
・カードゲーム用の拡張機能

といったものです。

グラフィック（ただしテキスト画面）のサポートはウィンドウエンジンの強化で行われる予定です。

コンパイルについてですが，基本部分はSX-BASICと同じですからＣ言語へのコンバートはさほど問題ないでしょう。BC.Xとフィルタをいくつか作ってＣ言語のライブラリを揃えればそれですむかもしれません。

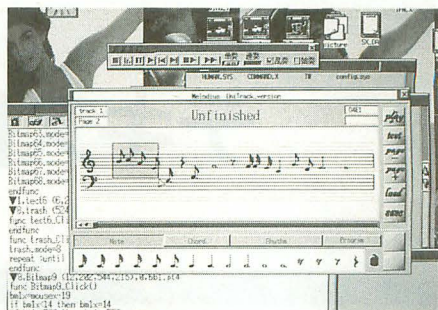
石上氏はコンパイラ作成のことも考慮しているようですので，そちらに期待するのもいいでしょう。

カードゲームはリソースへの対応というかたちで行われることになる予定です。SX開発キットの発売でようやくリソースの「安全な」使い方ができそうですので。

しかし現状の，アイテムの静的指定だけでは，たとえば「神経衰弱」をやる場合には54枚のカード分の関数を個別に書いてやらなければなりません。これだけで1000行くらいになるのではないでしょうか。根本的な部分でのシステム改善は必須といえるでしょう。

そのほか，私が個人的に要望しているのは，ひとつのウィンドウで複数のウィンドウスタイルを選択できる「ページという概念」です。たとえばページ１で設定されたウィンドウアイテムはページが切り替わると自動的に消去され，新しいページのアイテムが配置されます。こういったものがあればHyperCardのようなものも簡単に作成できるでしょう。

テキストも拡張フォーマットに対応させるべきかもしれません。現在はウィンドウエンジンに直接文字列90文字ごとに渡しひいますが，これをハンドル（要するに文字列の格納場所）渡しにすれば柔軟な処理ができ，かつ高速化できます。最大の問題はSX-BASICではハンドルの扱いが困難だということですが……(すでにputで実現している機能ですから，なんとかなるか？)。

操作性の点でいえば，エディタ部の改良

楽譜エディタ（開発中）

があります。せめてカーソル以降を消去することくらいはできたほうがいいのですが……。シャーペンの外部ファイルフォーマットがわかれば，SX-BASICと直接リンクするようなこともできるでしょう。

そのほか，タスク間通信でのタスクIDの検索や，ファイルメニューの装備など拡張すべきものは山積み状態です。

さすがにこれでは石上氏の負担が大きすぎますので，拡張法を明確にして，負担を分担できるようにすることが先決かもしれません。極端な話でいえば「配列をコールするような関数」とか，コードリソースを使って外部関数のようなものを作ることやウィンドウエンジン自体に柔軟な拡張性を持たせることなどです。

＊　＊　＊

とりあえず来月号には暫定版第２号が発表されますので，それを見てまた皆さんの意見を聞かせてください。投稿のほうもよろしくお願いします。

やはりオブジェクト指向か？

アイテムの動的扱いについては現在いろいろと検討されていますが、当分解決しそうにありません。

プロパティや関数を実行時に評価するようなものでは、コンパイラ作成の際にネックになることが予想されます。変数的に扱うにしてもメソッドやプロパティまわりの引数が不定個となっていますから、いろいろとやっかいなようです。このあたりの扱いを一元化できるように仕様を変えることも考えられます。

プログラムの感じとしてはワイルドカードのように指定するのがいちばん自然なので、暫定的にSX-BASICにプリプロセッサを内蔵して中間言語に落とす際に展開するとか、少々姑息な手段も考えられます。

しかし、一時的に問題を解決できても、将来的に禍根を残すことになるかもしれません。こういった問題ではちゃんとしたオブジェクト指向避けて通れない道なのでしょうか。そうなると処理系そのものの大幅な変更が必要になってくるでしょう。別のシステムにしたほうがよいのではなかという意見もあります。

オブジェクト指向に詳しい方、ちゃんと処理するためになにかいいアイデアはないでしょうか？

拡張テキストフォーマットを使う

テキストマネージャ解析結果

Tamura Kento 田村 健人

S X - W I N D O W

シャーペン.Xの登場から1年。自由自在な書式や書体指定で絶賛された新しいテキストマネージャの機能はついに公開されませんでした。ここでは謎に包まれていた拡張テキストフォーマットの解析結果を発表します。

皆さんはバナナの皮で転んだことがあるでしょうか? 私はありません。転んだことはありませんが,なぜか道端にバナナの皮が落ちているのを何度か見たことがあります。

バナナの皮を踏んで転ぶという漫画的な状況は起こり得るのでしょうか。少なくとも,前述のように道にバナナの皮が落ちていることがあります。ということは,本当にバナナの皮を踏んで転びやすいのかどうかを確かめればよいのです。私が小学生のころ,給食でバナナが出たときに試したことがあります。バナナの皮を踏むと,実際滑りやすいようです。

以上,自分で調べるということは大切であるという例です。そう,わからないからといって,わからないままにしてはいけないのです。そこで,私はテキストマネージャの解析をしました。

ところで,この記事はえらく敷居の高いものになっています。

・SX-WINDOWのテキストエディットの扱いがわかる

・gcc-SXの動作がわかる

・アセンブラを読める

以上の条件を満たしていないと,満足に読むことができないでしょう。いったい全国で何人の方がこの記事を理解できるのか非常に不安です。

この記事に書いてある情報はあくまで個人の解析に基づいて書かれています。私にも限界がありますので,おそらく間違った情報などもあるでしょう。くれぐれもシャープやF.C.T.,その他関係のありそうな団体へ問い合わせたりしないでください。また,Oh!X編集部へ問い合わせても答えられる人はいませんし,私個人に聞かれてもここに書いてある以上のことはわからないでしょう。

驚くほど多彩なテキストの機能

いきなり使ってみる

```
__asm ( "a470¥tequ $a470¥n" ) ;
__SXCALL long a470(rect*prDest,
rect*prView, graph*pg, short*ps) ;
```

gcc-SXの環境でプログラミングしている人は,上の宣言をどこかに入れて,従来TMNew() TMNew2() TMOpen() で作っていたテキストエディットをa470()で作ってください。

たとえば,

```
ht = TMNew2( &rDest, &rView, &
pwin->wGraph ) ;
```

としていたものは,

```
short s [6] = { 0,0,0,0,0,0, } ;
a470( &rDest, &rView, &pwin->
wGraph, s ) ;
ht = (tEdit**)_SXCALLPtr;
```

と変更します。

なんと,たったこれだけの変更で修飾付きテキストが扱えます。これ以外の部分は従来のテキストエディットとまったく同様に扱えるのです。さすがにこれだけでは修飾の変更などはできませんが,シャーペン.Xで入力した派手な文書をペーストすることができます。

実際にやってみた人は「あれ? 色が」とか「フォントサイズがぁぁ」というトラブルに遭遇するかもしれませんが,そのあたりはこの記事を読んでいけば解決することと思います。

アセンブラ派のための補足

gcc-SXでのSXコール呼び出しについて説明します。

引数は,shortならワード,short以外ならロングワードサイズとしてスタックに積みます。"型名*"はそのアドレスを表すので,ロングワードとなります。

__SXCALL宣言した関数での返り値は,d0レジスタです。また,呼び出しの直後の_SXCALLPtrがa0レジスタです。

●上記の a470 の例

```
s:  .dc.w   0,0,0,0,0,0
    ......
    pea     s(pc)
    move.l  pwin(a5),-(sp)
    pea     rView(pc)
    pea     rDest(pc)
    .dc.w   $a470
    lea     16(sp),sp
    move.l  a0,ht(a5)
```

シャーペンファイルの構造

シャーペンで保存したファイルをコマンドライン上で見て「なんやこの末尾のSTR@云々ちうのは」というのは多くの人が経験していると思います。このように,シャーペンではテキストの末尾に修飾の情報などを付加します。

末尾に付加するというのはテキストマネージャの仕様でそうなっているわけではなく,シャーペンの仕様です。昔の一太郎のように修飾の情報を別ファイルにして保存するということもやろうと思えばできることです。

表1がシャーペンファイルの構造です。

テキスト終端の^Zは次の'STR@'が偶数オフセットになるための調整を行います。テキストのサイズが奇数なら1個,偶数な

ら2個置かれます。

'STR@'で始まるのが修飾情報で,'EdEV'で始まるのがシャーペンの環境です。いちばん最後の1ロングワードが曲者で，この値が狂うとシャーペンで修飾と環境が認識されなくなります。コマンドライン上のエディタでシャーペンファイルを読み込んだ際に，STR@とEdEV中にあるコントロールコードが勝手に変換されてしまうと(たとえば，0x0dや0x0aが0x0d0aに変換されるなど),シャーペンに認識されなくなってしまうわけです。

STR@の構造

識別子'STR@'から始まる修飾情報は，タスクマネージャスクラップなどに収めるセルレコードそのものです。つまり，

```
long     識別子 'STR@'
long     以降のサイズ
任意長   情報本体
```

という形式になっています。

そして，情報本体は，

```
long     修飾が有効なテキストの長さ
可変長   修飾を表す情報
```

の繰り返しとなっています。この「修飾を表す情報」の構造体定義は次のようになります(文章がぶつぶつ切れて読みにくいですね。ごめんなさい)。

```
typedef struct {
    short fkind;
    short fface;
    short xfsize;
    short yfsize;
    long sizeadded;
    long added [0];
} str0;
```

なんとなくわかると思いますが，sizeaddedがaddedのバイト数を表しています。私はadded以降を「付加部」と呼んでいます。

sizeaddedが0で付加部がない場合,フォントカインド，フォントフェイス，フォントサイズのみを表します。

str0の頭に「修飾が有効なテキストの長さ」をくっつけてistr0を定義します。

```
typedef struct {
    long length;
    str0 s;
} istr0;
```

まとめると，STR@は，

```
'STR@', long, istr0, istr0, istr0 ....
```

または，

```
'STR@', long, long, str0, long, str0, long, str0, ....
```

という構造であるといえます。シャーペンでフォントをころころ変更するようなテキストを書いて，そのファイルをダンプしてみてください。ここに書いてあることがよくわかると思います。

str0の付加部について説明します。フォントカインド，フォントフェイス，フォントサイズ以外の情報はすべて付加部に格納されます。

```
long flags;
～可変長な中身～
long asize;
```

こういった形式になっています。flagsにより，中になんの情報が入っているかを示します。asizeは付加部全体のサイズで，flagsとasizeのぶんを含めたバイト数です。asizeは，この付加部が属するstr0のメンバsizeaddedと一致します。

flagsの構造を表2に示します。bit0～23は，そのビットが立っているときはサイズが「データ長」のデータが続きます。データはflagsの下位ビットにあたる項目から順に置かれます。bit24～31はなにか属性を表すもののようです。

bit8～15ではセルレコードがそのままデータとなります。

bit8はコマンドシェル上でテキストを見たときに「絵」となるもので，PAT1, PAT4, PICTなどのセルが置かれます。str0のフォントサイズがイメージの大きさと一致しています。

網掛けとルビではPICTが置かれます。

下線・中線・上線ではすべてPAT1が置かれます。シャーペンで作った線ではこのPAT1のバウンドレクタングルのleftが0x0000, rightが0x0010となっています。下線と上線ではtopが0x0000ですが，中線のみtopが負の値になります。

●センタリングされていて，文字間ピッチ4ドットの付加部

```
.dc.l   $00_00_00_11   *flags
.dc.w   1              *文字揃え
.dc.w   4              *文字間ピッチ
.dc.l   $0000000c      *asize
```

表1 シャーペンファイルの構造

長さ	内容
任意長	テキスト
1 or 2 バイト	^Z
long	'STR@' ┐
long	lenSTR@ ├ STR@
lenSTR@ バイト	修飾情報本体 ┘
long	'EdEV' ┐
long	lenEdEV ├ EdEV
lenEdEV バイト	環境情報本体 ┘
long	lenSTR@ + lenEdEV + 8

●0x200 バイトのPAT4の絵の付加部

```
.dc.l   $03_00_01_00   *flags
.dc.l   'PAT4'         *セルタイプ
.dc.l   $200           *セルサイズ
～ここに PAT4 そのものが入る～
.dc.l   $200+4+4+4+4   *asize
```

str0の付加部は可変長であるため，内容の操作が非常にやっかいです。特にC言語からの操作はかなり面倒です。そのためか，これを簡単に扱えるようにするため，付加部を展開したデータ構造が用意されています。

```
typedef struct {
    long flags;
    short sdata [8];
    cell* pcell [8];
    long ldata [8];
} expandedst;
```

flagsは付加部とまったく同一，sdata, ldataはデータそのもの，pdataはセルへのポインタです。また，この形式と付加部を相互に変換するSXコールが用意されています。

EdEVの構造

EdEVは修飾付きテキストエディットとは直接の関係はありません。この部分はシャーペン独自のものであり，これを扱うSXコールは用意されていないようです。しかしSTR@でサポートされていない有用な情報が多数含まれているので参照しないのは損です。

表3がEdEVの構造です。あまり本気で解析していないのでほとんどが不明です。

表2 付加部の flags

bit	データ長	意味
bit0	short	文字揃え
bit1	short	文字色－8
bit2	short	フォントモード
bit3	short	文字下ドット数＋0x4000
bit4	short	文字間ピッチ
bit5	short	強制改行幅
bit6,7	short	不明
bit8	任意	PAT4 ドロー
bit9	任意	網掛け
bit10	任意	下線 中線
bit11	任意	上線 ルビ
bit12～15	任意	不明
bit16～23	long	不明
bit24～31	0	0x03 PAT4 ドロー 0x40 網掛け 下線 中線 上線 0x4f ルビ 0x00 上記以外

テキストエディットレコード

TMOpen, TMNew, TMNew2で作られるテキストエディットレコードは，SX-WINDOWver.3.0 でもまったく変更されていないようです。a470で作られるものも，従来の上位互換となっています。拡張された部分を表4に示します。

テキストエディットレコードの大きさは0x9aバイトです。

オフセット0x24の欄が0なら従来のテキストエディット，0以外なら修飾付きテキストエディットです。

編集モードで改行コードの指定ができますが，これは修飾付きのテキストエディットだけに有効です。従来のテキストエディットでは利用できません。

プログラミングでの注意点

従来のテキストマネージャのコールがすべてそのまま利用できますので，テキストエディットの作成をa470で行ってしまえばあとは従来と変わりありません。

ひとつだけ注意しなければならないのは，a470をコールした時点でフォアグラウンドカラー，バックグラウンドカラー，アクセスページ，およびフォントの情報がテキストエディットに記憶されることです。

従来はTMEventなどを呼ぶたびにこれらの設定をやり直していたと思いますが，これがまったく必要なくなります。しかし，その代わりに，a470を呼ぶ前にこれらをきちんと設定しておく必要があるのです。気をつけてください。

最後に

修飾付きテキストエディットは，修飾を付けられることも特徴のひとつですが，ユーティリティコールの充実やフォント情報の保持により非常に扱いやすくなっています。改行幅が決まっていないと困るような状況以外では，積極的に使っていきましょう。

今回の解析作業を進めていくにあたって，dis ver.2.06βにはたいへんお世話になりました。作者のK.Abe氏に心より感謝いたします。

表3 EdEV の構造

offset	size	内容
+0x00	long	'EdEV': 識別子
+0x04	long	0x98： 総サイズ ここまでの8バイトを含まない
+0x08	long	'2.01': バージョン
+0x3c	short	bit0 改行文字表示 bit1 EOF表示 bit9 タブ表示
+0x40	short	スクロール行数
+0x4e	short	フォントフェイス
+0x50	short	フォントカインド
+0x52	short	1ラインドット数
+0x54	short	水平タブドット数
+0x5e	short	1： バックアップ作成
+0x60	short	最小改行幅
+0x64	short	2：シングルウィンドウモード
+0x66	short	1： 追い出し禁則モード
+0x6c	long	フォントサイズ
+0x72	short	アクセスページ
+0x74	short	1： オートインデントモード
+0x7c	short	禁則処理文字数
+0x7e	short	ワードラップ文字数

表4 テキストエディットレコード

offset	size	内容 [] 内はデフォルト値
+0x24	long	STR@ へのハンドル
+0x42	char	編集モード 改行コード指定 bit7 bit6 0 0： 0x0d0a 0 1： 0x0a 1 0： 0x0d
+0x72	long	初期サイズ 0x138 バイト のハンドル 現在の STR@ や改行幅テーブルが収められている
+0x76	short	フォアグラウンドカラー
+0x78	short	バックグラウンドカラー
+0x7a	short	フォントモード [2]
+0x7c	short	範囲選択時の色 [3]
+0x7e	char	最下位ビットにしか意味がないよう [1]
+0x7f	char	[不定]
+0x80	short	アクセスページ
+0x82	long	縦方向総ドット数 TMSetSelCal などで更新される
+0x86	long	ハンドル 詳細は不明だが，なぜか改行幅やタブ幅も入っている

リスト

```
  1: /*
  2:     SX-WINDOW v3.0 テキストマネージャ
  3:     gcc-SX 専用ヘッダ
  4:
  5:     このヘッダは自由に配布・改変してかまいません。
  6:     MethodSX や SXt に含めようがなんだろうが自由です。
  7:
  8:                                         24 Jan 94  by けんと
  9: */
 10:
 11: #ifndef     __TEXTV3_H
 12: #define     __TEXTV3_H
 13:
 14: #ifndef __SX_GCC__
 15: #error      このコンパイラでは利用できません
 16: #endif
 17:
 18: #ifndef     __SXDEF_H
 19: #define     _VARLEN 0
 20: #ifndef     __SXDEF2_H
 21: #include    <sxdef2.h>
 22: #endif
 23: #ifndef     __TEXT_H
 24: #include    <text.h>
 25: #endif
 26: #ifndef     __TASK_H
 27: #include    <task.h>
 28: #endif
 29: #define     __TEDIT TEdit
 30: #define     __TEHIS .EHis
 31: #define     __CELL  Cell
 32: #define     __GRAPH Graph
 33: #define     __POINT LPoint
 34: #define     __RECT  Rect
 35: #else
 36: #define     __TEDIT tEdit
 37: #define     __TEHIS teHis
 38: #define     __CELL  cell
 39: #define     __GRAPH graph
 40: #define     __POINT point_t
 41: #define     __RECT  rect
 42: #endif
 43:
 44: typedef struct {
 45:     short fkind;
 46:     short fface;
 47:     short xfsize;
 48:     short yfsize;
 49:     long sizeadded;
 50:     long added[0];
 51: } str0;
 52:
 53: typedef struct {
 54:     long length;
 55:     str0 s;
 56: } istr0;
 57:
 58: typedef struct {
 59:     long flags;
 60:     short sdata[8];
 61:     __CELL* pcell[8];
 62:     long ldata[8];
 63: } expandedst;
 64:
 65: typedef struct {
 66:     long kind;                  /* 'EdEV' */
 67:     long length;                /* 0x98 */
 68:     long version;               /* '2.01' */
 69:     short unknown0[24];
 70:     short bdisp;
 71:     short unknown1;
 72:     short nscroll;
 73:     short unknown2[6];
 74:     short fface;
 75:     short fkind;
 76:     short nwidth;
 77:     short ntab;
 78:     short unknown3[4];
 79:     short bback;
 80:     short nlf;
 81:     short unknown4;
 82:     short bsingle;
 83:     short boidasi;
 84:     short unknown5[2];
 85:     short xfsize;
 86:     short yfsize;
 87:     short unknown6;
 88:     short apage;
 89:     short bindent;
 90:     short unknown7[3];
 91:     short nkinsoku;
 92:     short nwrap;
 93:     short unknown8[16];
 94:     long sum[0];                /* おまけ */
 95: } edev;
 96:
 97:
 98:
 99: __asm (
100:     "a344   equ     $a344¥n"
101:     "a470   equ     $a470¥n"
102:     "a471   equ     $a471¥n"
```

```
103:    "a472   equ     $a472¥n"
104:    "a473   equ     $a473¥n"
105:    "a474   equ     $a474¥n"
106:    "a475   equ     $a475¥n"
107:    "a476   equ     $a476¥n"
108:    "a477   equ     $a477¥n"
109:    "a478   equ     $a478¥n"
110:    "a479   equ     $a479¥n"
111:    "a47a   equ     $a47a¥n"
112:    "a47b   equ     $a47b¥n"
113:    "a47c   equ     $a47c¥n"
114:    "a47d   equ     $a47d¥n"
115:    "a47e   equ     $a47e¥n"
116:    "a47f   equ     $a47f¥n"
117:    "a480   equ     $a480¥n"
118:    "a481   equ     $a481¥n"
119:    "a482   equ     $a482¥n"
120:    "a483   equ     $a483¥n"
121:    "a484   equ     $a484¥n"
122:    "a485   equ     $a485¥n"
123:    "a486   equ     $a486¥n"
124:    "a487   equ     $a487¥n"
125:    "a488   equ     $a488¥n"
126:    "a489   equ     $a489¥n"
127:    "a48a   equ     $a48a¥n"
128:    "a48b   equ     $a48b¥n"
129:    "a48c   equ     $a48c¥n"
130:    "a48d   equ     $a48d¥n"
131:    "a48e   equ     $a48e¥n"
132:    "a48f   equ     $a48f¥n"
133: );
134:
135: __SXCALL long a344( __TEDIT** ht, long x, long y );
136:                          /* 左上が (x,y) になるようにスクロール */
137: __SXCALL long a470( __RECT* prDest, __RECT* prView, __GRAPH* pg, str0* pint );
138:                          /* (__TEDIT**)_SXCALLPtr */
139: __SXCALL long a471( __TEDIT** ht, char** hText, long size, __CELL** hs, short fla
gs );
140:                          /* テキストと修飾をセット */
141: __SXCALL long a472( __TEDIT** ht, str0* ps );
142:                          /* これからの修飾を変更 */
143: __SXCALL long a473( __TEDIT** ht, istr0* pis, long offset );
144:                          /* 修飾情報を返す */
145: __SXCALL long a474( __TEDIT** ht, long, long, long, long );
146:
147: __SXCALL long a475( __TEDIT** ht, __CELL** hs, long offsrt_st, long offset_end );
148:                          /* 修飾情報を返す */
149: __SXCALL long a476( __TEDIT** ht, str0* ps, short mode );
150:                          /* 修飾を変更 */
151: __SXCALL long a477( __TEDIT** ht, short face, short mask );
152:                          /* fontface だけ変更 */
153: __SXCALL void a478( __TEDIT** ht, short fore, short back, short apage );
154:                          /* 各値を変更 */
155: __SXCALL void a479( __TEDIT** ht, short fmode, short fore2, short te7e );
156:                          /* 各値を変更 */
157: __SXCALL long a47a( long len, char** h, __CELL** hs );
158:                          /* テキストスクラップにコピー */
159: __SXCALL long a47b( __TEDIT** ht, char* pText, long size, __CELL** hs );
160:                          /* 修飾付き TMInsert */
161: __SXCALL long a47c( __TEDIT** ht, char* pText, long size, __TEHIS* pHis, __CELL**
 hs );
162:                          /* 修飾つき TMStr2 */
163: __SXCALL long a47d( __TEDIT** ht, __CELL** hs );
164:                          /* 修飾だけ変更する TMInsert */
165: __SXCALL long a47e( __TEDIT** ht, istr0* p0, long offset, expandedst* buf );
166:                          /* 付加部も含めて修飾情報を返す */
167: __SXCALL long a47f( void );
168:                          /* TMGetScrapLen */
169: __SXCALL long a480( __TEDIT** ht, long line );
170:                          /* 横ドット数 */
171: __SXCALL long a481( __TEDIT** ht, long line );
172:                          /* 改行幅 */
173: __SXCALL long a482( __TEDIT** ht, long line );
174:                          /* 総ドット数 */
175: __SXCALL long a483( __TEDIT** ht, long y );
176:                          /* 何行目 */
177: __SXCALL long a484( __TEDIT** ht, str0* p0, short mask, long mode2 );
178:                          /* 修飾を変更 */
179: __SXCALL long a485( expandedst* buf, istr0* pis );
180:                          /* 修飾を展開 */
181: __SXCALL void a486( __TEDIT** ht, short drawmode );
182:                          /* 編集モードを変更 */
183: __SXCALL long a487( void** hd, __CELL** hs, long flags, __POINT ptSize );
184:                          /* イメージを含む修飾を作る */
185: __SXCALL long a488( __TEDIT** ht, long foo );
186:
187: __SXCALL long a489( void );          /* 引き数不明 */
188:
189: __SXCALL long a48a( expandedst* buf, istr0* pis );
190:                          /* 修飾を変換 */
191: __SXCALL long a48b( __TEDIT** ht, short lineheight );
192:                          /* 最小改行幅を変更 */
193: __SXCALL long a48c( __TEDIT** ht, short tabsize );
194:                          /* タブ幅を変更 */
195: __SXCALL long a48d( __TEDIT** ht, char* ptext, long offset_st, long offset_end );
196:                          /* 範囲指定TMGetText (char*)_SXCALLPtr */
197: __SXCALL long a48e( __TEDIT** ht, __POINT pt );
198:
199: __SXCALL long a48f( __TEDIT** ht, __POINT pt );
200:
201: __SXCALL long a490( __TEDIT** ht, istr0* pis );
202:                          /* 修飾情報を返す */
203: __SXCALL long a491( void );          /* 引き数不明 */
204:
205: __SXCALL long a492( __TEDIT** ht, long foo, long bar );
206:
207: __SXCALL long a493( __TEDIT** ht, void** h, long foo, void* bar );
208:
209:
210: static __inline __TEDIT** a470_in( __RECT* prDest, __RECT* prView, __GRAPH* pg, s
tr0* pstr0 ) {
211:     a470( prDest, prView, pg, pstr0 );
212:     return (__TEDIT**)_SXCALLPtr;
213: }                        /* #define a470_in(A,B,C,D) (a470(A,B,C,D),_SXCALLPtr)
214:                           でもいいわけですが、型チェックをきちんとしたいから */
215:
216:
217: #undef      __TEDIT
218: #undef      __TEHIS
219: #undef      __CELL
220: #undef      __GRAPH
221: #undef      __POINT
222: #undef      __RECT
223:
224: #endif                        /* __TEXTV3_H */
```

修飾付きテキスト関連コール解析リファレンス

●リファレンスの読み方

返り値はd0.l, _SXCALLPtrはa0.lと読み換えてください。shortの引数はワードでスタックに積みます。各コールはCのプロトタイプのような形式で書かれています。返り値の型がないものは，未調査です。

_TMEventW equ $a46e

_TMUpDateExist equ $a46f

この2つはSX-WINDOW ver.2.0で追加されたもよう。詳しくは「追補版SX-WINDOWプログラミング」を参照してください。

long a470(rect* prDest, rect* prView, graph* pg, str0* ps);

返り値

リザルトコード。

(tEdit**)_SXCALLPtr;

機能

psで示される修飾情報を初期値として修飾付きのテキストエディットを作成する。

改行幅，タブ幅はpsで指定したフォントの高さとなる。

グラフポートのforeground color, background color, access pageがテキストエディットに記憶される。

long a471(tEdit ht, char** hText,long size, cell** hStr_at, short flags);**

返り値

リザルトコード。

機能

すでに作成してある修飾付きテキストエディットに対して，修飾付きのテキストをセットする。従来のテキストエディットでもいえることだが，テキスト終端の^Zを含めないサイズを渡したほうがよい。

hText	plain textを収めたハンドル。
size	textの大きさ。ハンドルの大きさではない。
hStrat	'STR@'以降を収めたハンドル。
flags	ハンドル2つをコピーするかどうかを指定する。上位バイトがhText, 下位バイトがhStrat。各バイトが0のときコピーしない。コピーしてもらうときは両ハンドルとも疑似ハンドルでよい。

a472(tEdit ht, str0* ps);**

機能

これから入力される文字の修飾を変更する。

long a473(tEdit ht, istr0* pis, long offset);**

long a490(tEdit ht, istr0* pis);**

返り値

指定した文字以降何バイト拘束するか。

(istr0*)_SXCALLPtr; もとにした修飾情報のアドレス。

機能

pisで指す領域に，offsetで指定した文字の修飾情報を返す。a490はoffsetがテキスト終端であるので，これから入力される文字の修飾情報である。str0の付加部は無視される。

longの部分には，offsetの文字以降，この修飾情報が何バイト有効かを示す。

もしstr0の付加部のbit16が立っていたら，str_at.sizeに，ロングワードデータが入る。

再配置は起きないので，_SXCALLPtrがそのまま利用できる。

a474(tEdit ht, long, long, long, long);**

a47c(修飾付きTMStr2)の下位コール。一般のプログラマが利用する価値はないと思われる。

a475(tEdit ht, cell** hStr_at, long offsrt_st, long offset_end);**

返り値

リザルトコード。

(void**)_SXCALLPtr;

機能

hStr_atのハンドルに指定範囲の修飾情報を返す。この内容は'STR@'から始まる。スクラップにコピーするときなどに使えるのだろう。

hStr_atは疑似ハンドル不可。hstr_at==NULLのときは新規にハンドルを作成する。このハンドルはユーザーが破棄すること。

a476(tEdit ht, str0* ps, short**

mask)；

機能

修飾をセットする。範囲選択されているときはその範囲，されていないときはこれから入力される文字の修飾となる。

psの先に修飾情報を書いておく。

内部でa484を呼んでいる。

mask:

bit 0　フォントカインドを設定する
bit 1　フォントフェイスを設定する
bit 2　フォントの横サイズを設定する
bit 3　フォントの縦サイズを設定する

すべてのビットが0のときは，付加部のみ設定される。

a477(tEdit** ht, short face, short mask)；

機能

フォントフェイスだけを変更する。

mask:

bit 0　強調にするかどうかを変更する
bit 1　イタリックにするかどうかを変更する
bit 2　アンダーラインを付けるかどうかを変更する
bit 3　中抜きにするかどうかを変更する
bit 4　影付きにするかどうかを変更する

void a478(tEdit** ht, short fore, short back, short apage)；

機能

foreground color,background color,access pageを設定する。再表示などは行わない。ただ値をテキストエディットレコードに書き込むだけである。

void a479(tEdit** ht, short fmode, short fore2, short te7e)；

機能

フォントモード，反転時の色を設定する。再表示は行わない。

te7eはテキストエディットレコードのオフセット0x7eに書き込まれる。申し訳ないがこの部分の用途がまだ不明なので，引数も不明である。通常は初期値である0x0100としておけば安全だろう。

long a47a(long len, char** h, cell** hs)；

返り値

リザルトコード。

機能

h内の長さlenの文字列と，修飾情報hsをテキストマネージャスクラップにコピーする。両方とも疑似ハンドル可。hsの中身は'STR@'から始まる。

hs==NULLのときは，文字列のみがコピーされる。

このコールの使用直後にTMToScrap()をしないとほとんど意味がないと思われる。

long a47b(tEdit** ht, char* pText, long size, cell** hs)；

返り値

リザルトコード。

機能

修飾付きのTMInsert。

hsは'STR@'で始まる。hs==NULLのときTMInsertとまったく同一。

a47c(tEdit** ht, char* pText, long size, teHis* pHis, cell** hs)；

機能

修飾付きのTMStr2。

hsは'STR@'で始まる。hs==NULLのときTMStr2とまったく同一。

a47d(tEdit** ht, cell** hs)；

機能

修飾だけ変更するTMInsert。

a47b(ht, NULL, -1, hs)；としてからごにょごにょしている。

a47e(tEdit** ht, istr0* p0, long offset, expandedst* buf)；

返り値

(istr0*)_SXCALLPtr;

もとにした修飾情報のアドレス。

機能

offsetの位置の修飾情報を返す。a473では付加部の情報が得られなかったが，このコールでは得ることができる。expandedstのセルへのポインタは再配置が発生した時点で無効となる。

long a47f(void)；

機能

a328 TMGetScrapLenとまったく同一。なんの意味があるのだろう？

long a480(tEdit** ht, long line)；

返り値

line行目の横ドット数。

機能

line行目の横幅を求める。

以下3コールは，lineを負にすると0行目，最大より大きくすると最後の行となる。

long a481(tEdit** ht, long line)；

返り値

line行目の改行幅。

long a482(tEdit** ht, long line)；

返り値

line行目までの縦方向総ドット数。

long a483(tEdit** ht, long height)；

返り値

height ドットの座標が何行目に相当するか。

a484(tEdit** ht, str0* p0, short mask, long mode2)；

機能

a476と同一。引数の与え方が異なるだけである。

maskはa476のmaskと同じ。

付加部がない場合にはmode2=0，ある場合には付加部のflagsを与える。

a476の下位コール。

long a485(expandedst* buf, istr0* pis)；

返り値

付加部分の大きさ。

(istr0*)_SXCALLPtr　次のistr0

機能

pisで示される修飾情報をbufに展開する。expandedstのセルへのポインタは再配置が発生した時点で無効となる。

再配置は起きない。

void a486(tEdit** ht, short drawmode)；

機能

編集モードの値をdrawmodeに変更する。再描画されない。

long a487(void** hd, cell** hs, long flags, point_t ptSize)；

返り値

リザルトコード。

(void**)_SXCALLPtr;

機能

hdをもとにhs内に可変長データ付きSTR@を作る。hsには'STR@'も含まれる。hsは疑似ハンドル不可。hs==NULLのときは新たにハンドルを確保する。

PAT4を作る場合は，flags=0x03000100，hdには，

'PAT4'
以降サイズ
PAT4 そのもの

を入れておく。

このコールのあとにa47bを呼べばイメージペーストができるのだろう。

a488(tEdit** ht, long foo)；

不明。

a489

不明。a493を呼んでいる。

a48a(expandedst* buf, istr0* pis)；

機能

a485の逆。bufに展開されている修飾情報をpisに収める。pisの領域の必要量はbufの中身からわかるはず。

long a48b(tEdit** ht, short lineheight)；

返り値

リザルトコード。

機能

最小改行幅をlineheightドットに設定する。修飾付きテキストでなくても可。値をテキストエディットレコードに書き込むだけで，再描画などは行わない。

long a48c(tEdit** ht, short tabsize)；

返り値

リザルトコード。

機能

タブ幅をtabsizeドットに設定する。修飾付きテキストでなくても可。あとは同上。

long a48d(tEdit** ht, char* ptext, long offset_st, long offset_end)；

返り値

文字数。

(char*)_SXCALLPtr　文字列の終端+1

機能

offset_stからoffset_endまでのテキストをptextに返す。修飾付きテキストエディットでなくても利用可。

long a48e(tEdit** ht, point_t pt)；
long a48f(tEdit** ht, point_t pt)；

不明。なにかを計算して返している。テキストエディットの内容はまったく影響を受けない。

a490

→a473

a491

不明。文字を実際に描画するためのコールのようだ。

long a492(tEdit** ht, long foo, long bar)；

不明。

a493(tEdit** ht, void** h, long foo, void* bar)；

不明。文字を描く？

こちらシステムX探偵事務所

FILE-XI

ボールを動かす

Shibata Atsushi 柴田 淳

ピンボール作成の2回目です。今月はピンボールの最も基本的な部分「ボールの動き」がテーマです。いかにして滑らかにボールを動かすのか。どのようなアプローチをしていくのでしょうか。見逃さないようにしてください。

illustration：T.Takahashi

小さかった頃，僕は「Bブロック」というブロックで遊んでいた。ブロックというとLEGOのような小さいものを思い浮かべる人が多いかと思うが，Bブロックはもっと大きなブロックだった。子供の手のひらにすっぽり収まるほどの大きさ。ブロックには親指よりひと回りほど大きい突起が2つついており，反対側には突起がはめ込まれる穴が開いている。ブロックの側面は片方だけ角張っていて，もう一方は丸みを帯びている。つまり，突起をはめ込む穴の開いている側からブロックを見ると，ちょうど「B」の形に見えるわけである。これが「Bブロック」と呼ばれる所以だ。

LEGOなどのブロックには，長いパーツや窓のパーツなど，さまざまな形状があって，これを組み合わせることによって比較的リアルなものを組み上げることができる。しかし，僕の持っていたBブロックには，B型のブロック，L字型の両端にひとつずつの突起のついたもの，ブロックの両面が「雌」であるブロックなど，割に単純なパーツしか用意されていなかった。で，単純に考えると，BブロックはLEGOなどよりパーツの種類が少ない分，面白味に欠けると思うだろうが，実は違うのである。

Bブロックの楽しさの秘密は，その独特なB型の形状にある。たとえば，横に4つの突起が並んだLEGOのパーツを使い，輪を作るとする。LEGOは側面が角張っているので，ブロックを隣り合わせるとき，角度をもたせることは難しい。また，ブロックを煉瓦のようにハスに積み上げていっても，やはり角がブロックの突起に当たってしまい，大きな角度をとれない。ところがBブロックの場合は，Bの丸まった腹の面を輪の内側に向けることで，隣り合ったブロック同士どんな角度でもとらせることができる。また，Bブロックのこの性質を応用すると，可動の物体を組み上げることができる。動かしたい部分，つまり関節にあたるところはBの丸まった腹を同方向に向ける。逆に動かしたくない部分は，腹の面を互い違いに組めばいい。

普通，児童心理学では，ブロックは「子供の構成力を育てる遊び道具」とされている。Bブロックも，僕の構成力を育てる役にはたったはずだ。しかし，ブロック遊びをしていた頃を思い出すたびに，Bブロックは構成力以上に「論理性」を培ってくれたような気がする。

Bブロックはパーツの種類の割には自由度の高いブロックだが，その分複雑な形を組もうとすると苦労が伴う。特にL字型のブロックなどは数が限られているので，ほかのブロックで代用しなければならないときが多い。代用といっても，ブロックの並べ方によっては不可能な組み合わせも出てくる。つまり，ブロックの組み方には論理的な制約があるのだ。思うような形を組み上げるために必要なのは，この論理を踏まえた綿密な計画と経験である。この種の体験は，お城セットを買ってきたらお城しか組めないLEGOのようなブロックでは，決して味わうことはできなかっただろう。

限られたパーツだけを使い，失敗しても根気よく，目的の形を作っていく。子供の頃，幾度となく繰り返したこの「遊び」と，遊びながら身につけた「論理性」は，現在プログラミングと結びついている。小さい頃Bブロックで遊んでいなかったら，僕がプログラミングという道に踏み出すことはなかっただろう。

滑らかさについて考える

「論理」というのは「抽象的思考」にほかならないのだが，生来から抽象的思考に動機づけられている人間はいない。人は論理を，生まれてからの学習によって修得するのである。

コンピュータのプログラミングにはかなりの論理性を要求される。ということは抽象的思考法を学んでいるかいないかが，プログラミングに関する適性にかなり影響を与えるはずだ。実際，遺伝的性質，発育環境ともに僕と似ている年子の弟は，コンピュータには興味があっても，プログラミングにはまったく疎い。小さい頃，ブロックを兄に独占されていたからだろう。

プログラミングをするときに要求される類の論理性というのは，抽象的思考に慣れていない人からすれば拷問に近いものだろう。コンピュータというのは，頭がいいようでいてその実，世間知らずな機械である。たとえば，人間だったら「歩け」といわれれば歩くことができる。しかし，コンピュータの場合はそうはいかない。「歩くとは脚を交互に前に出すことだ」とか，「脚を前に出すためには腿を上げて……」「だめだめ，脚を動かすときはバランスを考えなくちゃ」などと，いちいち教え込まなければならない。

そして，ピンボールゲームを作るとなると，ボールの動きを「論理的に記述」しなければならないのだ。ボールと壁との反射に関してはすでに先月号で終えているが，これでもボールの挙動のうちのほんの一部を記述したにすぎない。ボールの動くベクトルと当たる壁の角度がわかっているときに，反射したあとの方向ベクトルを得られても，ボールがどのように動き，いつ接触したかがわからなければ意味がない。

目指すところは，ボールを本物らしく動かすことである。この「本物らしく」を念頭に置くと，「ボールを滑らかに動かす」ということは重要な要素だ。

さて，ご存じの通り，テレビの画面は60分の1秒ごとに描き換えられている（垂直表示周期）。ボールの表示されている位置を

少しずつ動かし，ボールを動いているように見せることができるのはいうまでもない。逆にいえば，ボールの位置をどんなに速く動かそうとしても，描き換え速度は60分の１秒の周期より短くはならないということだ。で，ボールを滑らかに動かすためには，60分の１秒に１ドットずつ動かせばいいのではないかと考えたくなる。これで確かにボールは滑らかに動くが，速度が遅すぎる。256ドットを移動するのに，4秒ほどかかる計算になる。

ボールは壁との跳ね返り具合によって，さまざまな速度をとるわけだ。そして画面の垂直表示周期は動かせないのだから，ボールの移動距離で速度を表現しなければならない。速度が遅いときは60分の１秒の間に１ドットも動かない場合もあるだろう。また，周期中に数十ドット動かさなければならないときもあるかもしれない。

画面上の物体を動かす距離をいろいろ変えて実験してみた結果，60分の１秒に20ドット動かしてもまだ動きは滑らかに見えた。これは,256ドットを約0.21秒で移動する速度だ。物体は実際かなり飛び飛びに表示されているのだが，スピード感が増しているので，人間の目には滑らかに動いていると認識されるようである。

つまるところ，滑らかさというのは動いている物体の速度などの要素に依存する，ということだろうか。すると，ボールを動かすときに当面必要なのは，速度に合わせてボールの表示する座標を移動させるような手法である。反射の処理の際にボールの移動方向を表すベクトルが必要なのだから，このベクトルをボールの座標移動に使えれば都合がいい。また，ボールが特に遅いときは，１ドットずつの移動すらしない場合があるだろう。それでいて，ボールは動いているように見えなければならないのだ。

これらの条件を満たすオーソドックスな手法として，固定小数点というものがある。普通4バイトで表現される整数を，下位の2バイトを小数部，上位の2バイトを整数部に見立てる，という手法だ。ボールの移動方向を示すベクトル，つまりＸ軸とＹ軸方向の移動量を，固定小数点を使って用意する。この移動量をボールの座標に足すと，整数部だけが座標移動として表示に反映される。ボールの表示位置からは小数部は見えてこないが，実際には座標に足されている。したがって何回か足し算を繰り返すうち，繰り上がって座標の整数部に影響を与えるわけだ。

画面の表示というのは，実際は60分の１秒という単位で区切られたデジタルな世界である。この世界で物体を動かそうとすると，現実世界のアナログ的な手法とは違った方法論が必要になってくる。すると当然,「滑らかさ」もデジタルの洗礼を受けなければならない。たとえば，平らな面の上で物体を一定速度で動かすとする。移動は連続であり，計測時間の間隔をどんなに細かくとっても移動する距離を測定できる。簡潔にいえば「微分可能」なわけである。

ところが，モニタ上で物体を動かすとなると状況は変わってくる。まず計測時間は，常に60分の１秒の倍数になる。物体の移動は不連続で「微分不可能」である。このような性質をもった世界を取り扱うために，デジタル微分解析という数学の分野がある。ここでボールの移動に使った手法もこれを応用したものなのだが，アクション性のあるゲームを作りたいなら，なんらかの形でデジタル微分解析を学ぶことになるだろう。

ボール移動の実際

さて，これでボールを表示する位置を刻々と変化させ，60分の１秒ごとのボールの座標を決めることができるようになった。あとは，表示されているボールが壁に接触するような位置にあるとき，前回の反射の公式を使って反射後のベクトルを求めればいいはずなのだが，実際はそう簡単にことは運ばない。

試しにこういう場合を想定してみよう。ボールの移動量は，ボールの速度が速いときは数ドットに達することがあると前述した。では，あるときボールが壁に接触する直前の位置にあり，そこから壁に向かって数ドット移動したとする。すると，ボールは壁にめり込んでしまうことになる。めり込んだ場合でも壁とボールが接触していることには変わりがないので，プログラム中では反射の処理がなされるだろう。しかし，ボールが壁を完全に通り越してしまう，という最悪の場合も想像できる（図1）。

壁との反射の処理を行うのは，ボールが壁に接触（衝突）した瞬間でないと都合が悪い。要するに，ボールの座標をいっぺんに数ドット動かすと不都合が出てくるわけだ。固定小数点という手法を導入することによって，ボールの移動速度に関係なく表示座標を特定できるようになったのに，これでは問題が逆戻りしてしまう。

ここで頭を使う。要は，ボールを画面に表示する場所を，移動方向のベクトルによって決められた座標に取ればいいのである。もっというと，ボールが新しい位置に表示されるまでの60分の１秒の間は，ボールの座標がどこにあろうと関係がないわけだ。またボールが壁に接触した瞬間を検出するには，ボールを１ドットずつ移動させればいい。１ドットずつの移動を繰り返し，ボールを本来移動すべき座標までもっていく。この作業を，60分の１秒の間に行い，新しい位置にボールを表示すれば，見かけ上はボールをいっぺんに数ドット動かしたときと変わらない。

１回の移動でボールの座標がどれだけ移動するかは，固定小数点で表現された移動方向を表すベクトルの整数部を見ればわかる。リストに目を移すと，ボールの座標を動かしているmove_ballという関数内の125，126行で，移動方向ベクトルから整数部だけを取り出しているのがわかるだろう。固定小数点では，4バイトのうち上位2バイトが整数部に当たるから，ベクトルのＸ成分，Ｙ成分をそれぞれ16ビット分右にシフトすればいいわけだ。

こうして得られた数値をもとに，ボールを１ドットずつ動かすために必要なパラメータを算出しているのが,172行からのcalc_parm関数である。まずボールの移動量のＸとＹのどちらか絶対値の大きいほうを，構造体r_parmのメンバｆに渡す。また，構造体r_parmにはＸ方向とＹ方向用に使うためのカウンタが用意されている。このカウンタにボールの移動量を足していき，Ｘ，Ｙのカウンタがメンバｆの値を越えたら座標を増減させる。リスト中では，138行からのdo～whileループがその処理をしている部分だ。

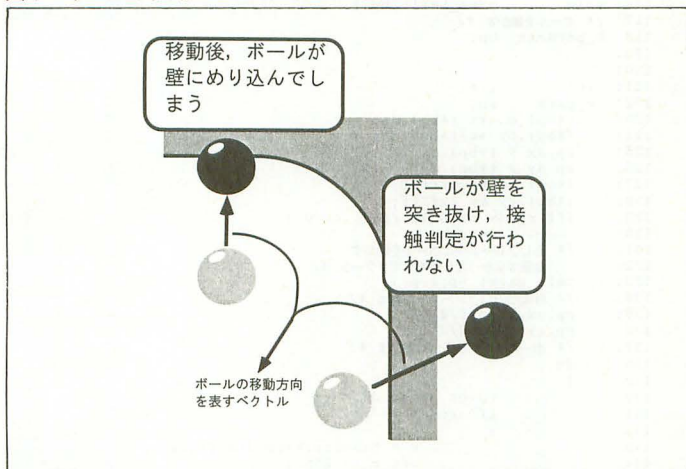
つまり，ビットマップスクリーンに線を引くときの要領で，ボールの座標を動かしていくのである。こうすれば，ボールの通る軌跡をくまなく埋め，なおかつ表面上は，移動方向ベクトルを用いてボールを移動させたときと変わらないように見せることができる。

あとは，この方法でボールの座標を移動

させていき，座標が1ドット動くたびに壁との接触判定を行う。そして，壁と接触している場合は反射の処理をするのだが，実はここにも落とし穴が待ちかまえている。

反射後の処理

ボールが壁に接触したことがわかると，反射の処理によってボールの方向を表すベクトルが変わる。すると，ボールの座標を1ドットずつ動かすときに使う，パラメータの値も変えなければならない。ここで，横着をしてボールの座標移動を止めてしまうと問題が起こる。

図2を見てほしい。ボールの表示位置が壁の近くにあるときに，壁と接触した時点でボールを止めた場合，この図のようにボールが本来移動すべき距離だけ動ききらない。この現象が具体的にどのようなものか，残念ながら説明することは難しい。なにしろ60分の1秒というごく短い時間に起こる事柄である。たんにボールが壁に当たって跳ね返る場合,「少し止まったような気がする」。

止まったような気がするだけならたいしたことはないが，ボールが壁伝いに動く場合，問題はより深刻になる。壁伝いのボールの運動は，小さな反射の繰り返しとして表現される，ということについては先月号で触れた。で，反射の際にボールを止めてしまうと，本来60分の1秒の間に行われるべき反射の処理が行われず，ボールは曲がり口にさしかかったところで止まってしまう。止まった先にも壁があるので，次のボールの移動も壁に行く手を阻まれる。結果，ボールの移動速度は急激に遅くなってしまう。reflect関数中の245行から261行をコメントにするなどして削除し，リストをコンパイルすると，この現象を見ることができる（図3左）。

では，反射後もボールの座標を動かすためにはどうすればいいだろうか。まず，本来移動させるべき距離のうち，壁に接触した時点でどれだけボールが動いているかを調べる。すでに動いた距離を図4のようにMとし，切り捨てられてしまう部分をNとしよう。また，先月号の公式を使って求めた反射後のボールの移動方向ベクトルを$\vec{R}$とすると，簡単な比率の計算を行い，図4の式を得る。あとは得られた数値をもとにして新しいパラメータを算出し，ボールを移動させればいい。

反射の際にこのような処理を行うことで，図3の右側のようにボールを壁伝いに動かすことができるようになる。ここで取り上げた問題も，ボールの動きを表現する世界がデジタルであるがゆえに起こる現象だといえる。

サンプルについて

さて，先月号と今月号の記事内容を検証するためのサンプルプログラムについて少々。このプログラムをコンパイル，実行すると，画面に角の丸まった四角形が表示される。ボールをマウスで好きなところに動かし，左クリックするとボールが落下し，壁との衝突などを繰り返し跳ね回る。

ボールが動いているとき，右クリックでもう一度ボールを置くところから始めることができる。プログラムから抜けるときは，ボールが動いているときに左クリック。

図1　ボールの移動量を速度で決めた場合

図2　壁と接触した時点でボールを止めた場合

図3　ボールが壁伝いに動く場合

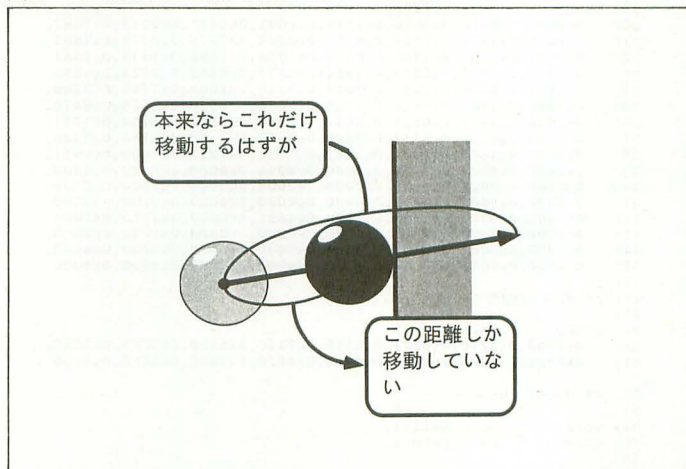
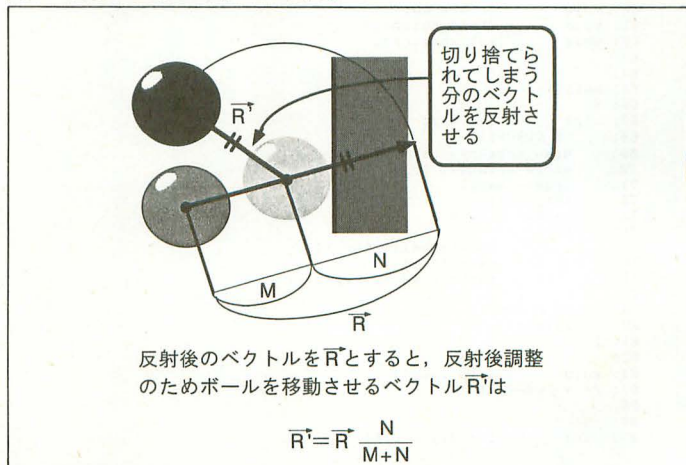



図4　反射したあとのボールの移動

実際のピンボール台のように，地面に対し傾いた台の上のボールの挙動をシミュレートすることを念頭に置いて，引力の作用は弱めに設定してある。したがってあまり派手な動きは見せないが，それなりに本物らしく見えるはずだ。また，特に工夫もなくCで書いているプログラムでも，ボールの動きが遅くなることはない。だが考えてみれば，プログラム中でしていることといえば，たかだか数回の足し算や掛け算である。そこそこの速度を得られるのも，当然かもしれない。

本当は今月号でパドルとボールの反射も説明しようと思っていたのだが，そこまでたどり着けなかった。実際プログラムを組んでみると，ボールの挙動を再現するというのは，考えていた以上に骨の折れる作業であった。

また今回のサンプルプログラムには，反射時のボール速度の減衰や，接触判定の簡略化など，まだ説明していない手法も盛り込まれている。それに今回のサンプルでは，ボールの動きに怪しい部分が見られる。基本的な手法の変更はしないだろうが，アルゴリズムなどの微調整を試みつつ，もうしばらくボールの動きにこだわってみたい。

なお，来月号には付録ディスクがつくということなので，もう少し大きな画面上をボールが動き，パドルのついたサンプルを収録していただくつもりでいる。(つづく)

リスト

```
  1: /*
  2:     ボールの動きをテストする
  3:
  4:          1994.2.6 (日) (ATS)   */
  5:
  6: #include "stdio.h"
  7: #include "iocslib.h"
  8: #include "basic.h"
  9: #include "graph.h"
 10:
 11: typedef    struct  b_parm {
 12:    int     x,y;
 13:    int     dx,dy,cx,cy;
 14:    int     drx,dry;
 15:    } b_parm,*b_parmPtr;
 16: /* ボールのパラメーター構造体 */
 17:
 18: typedef    struct  r_parm {
 19:    int     x,y,dx,dy,cx,cy,f,drx,dry;
 20:    int     dxx,dyy;
 21:    } r_parm,*r_parmPtr;
 22:
 23: b_parm     bp;
 24:
 25: UBYTE      bmap[256][256];
 26:
 27: int        si[64],co[64];
 28:
 29: UWORD      ballpat[2][64] = {
 30:   0x0000,0x0012,0x0000,0x1246,0x0001,0x3677,0x0013,0x7887,
 31:   0x0036,0x8877,0x0147,0x8776,0x0268,0x7776,0x0378,0x7665,
 32:   0x1477,0x6557,0x1456,0x6578,0x1356,0x5686,0x1445,0x5554,
 33:   0x0344,0x5445,0x0235,0x4354,0x0135,0x6555,0x0025,0x5655,
 34:   0x1100,0x0000,0x2531,0x0000,0x5348,0x1000,0x7366,0x3200,
 35:   0x6546,0x5800,0x5646,0x5720,0x5656,0x6520,0x8555,0x6470,
 36:   0x7645,0x6672,0x6535,0x5671,0x5544,0x5681,0x5454,0x5771,
 37:   0x4345,0x5730,0x3443,0x5520,0x5665,0x3410,0x6786,0x4300,
 38:   0x0013,0x4666,0x0001,0x3487,0x0000,0x1222,0x0000,0x0011,
 39:   0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
 40:   0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
 41:   0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
 42:   0x7885,0x3100,0x8863,0x1000,0x4521,0x0000,0x1100,0x0000,
 43:   0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
 44:   0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
 45:   0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000
 46: };
 47: /* ボールのスプライトパターン */
 48:
 49: UWORD      pal[16] = {
 50:   0x0000,0x4210,0x5294,0x6318,0x739C,0x8420,0x9CE6,0xB5AC,
 51:   0xFFFE,0x4000,0xCE72,0x4010,0xDEF6,0x2D8C,0x5ECB,0x9FD5 }
;
 52: /* ボールのパレット */
 53:
 54: void       move_ball();
 55: void       calc_parm();
 56:
 57: void       locate_ball();
 58: void       put_ball();
 59: void       draw_screen();
 60: void       create_map();
 61: void       store_sctab();
 62: void       set_sprite();
 63:
 64:
 65: main()
 66: {
 67:  int       i = 0,j;
 68:    OS_CUROF();
 69:    draw_screen();
 70:    store_sctab();
 71:    create_map();
 72:    set_sprite();
 73:    while( i == 0 )
 74:    {
 75:            locate_ball( &bp );
 76:            main_loop();
 77:            msstat( &j,&j,&i,&j );
 78:    }
 79:    OS_CURON();
 80:    exit(0);
 81: }
 82:
 83: void       main_loop()
 84: /* メインループ */
 85:
 86:  {
 87:  int       i = 0,j = 0,k;
 88:    /* ボールのパラメーター構造体の初期化 */
 89:    bp.dx = 0;
 90:    bp.dy = 0;
 91:    bp.drx = 0;
 92:    bp.dry = 0;
 93:    bp.cx = 0;
 94:    bp.cy = 0;
 95:    while( i == 0 && j == 0 )
 96:    {
 97:            /* 引力の処理 */
 98:            if( bp.dry == 1 )
 99:                    bp.dy += 0x1000;
100:            else
101:            {
102:                    bp.dy -= 0x1000;
103:                    if( bp.dy < 0 )
104:                    {
105:                            bp.dy = -bp.dy;
106:                            bp.dry = 1;
107:                    }
108:            }
109:            /* ボールを動かす */
110:            move_ball( &bp );
111:            put_ball( bp.x,bp.y );
112:            msstat( &k,&k,&i,&j );
113:    }
114:  }
115:
116: void       move_ball( bp )
117: /* ボールを動かす */
118: b_parmPtr  bp;
119:
120:  {
121:  int       i,r;
122:  r_parm    rp;
123:    (*bp).cx += (*bp).dx;
124:    (*bp).cy += (*bp).dy;
125:    rp.dx = (*bp).cx >> 16;
126:    rp.dy = (*bp).cy >> 16;
127:    (*bp).cx &= 0xffff;
128:    (*bp).cy &= 0xffff;
129:    if( rp.dx == 0 && rp.dy == 0 )
130:            return;
131:    /* もしフレーム中にボールを動かす
132:       必要がなかったら何もせずリターン */
133:    calc_parm( bp,&rp );
134:    /* 移動用のパラメーターを算出 */
135:    rp.cx = rp.f/2;
136:    rp.cy = rp.f/2;
137:    /* ボールを1ドットづつ動かす */
138:    do
139:    {
140:            rp.cx += rp.dx;
141:            if( rp.cx >= rp.f )
142:            {
143:                    r = bmap[rp.x+rp.drx][rp.y];
144:                    if( r != 255 )
145:                            reflect( bp,&rp,r );
146:                    else
147:                    {
148:                            rp.dx--;
149:                            rp.x += rp.drx;
150:                    }
151:                    rp.cx -= rp.f;
152:            }
153:            rp.cy += rp.dy;
154:            if( rp.cy >= rp.f )
155:            {
156:                    r = bmap[rp.x][rp.y+rp.dry];
157:                    if( r != 255 )
158:                            reflect( bp,&rp,r );
159:                    else
160:                    {
161:                            rp.dy--;
162:                            rp.y += rp.dry;
163:                    }
164:                    rp.cy -= rp.f;
165:            }
166:    }
167:    while( rp.dx > 0 || rp.dy > 0 );
168:    (*bp).x = rp.x;
169:    (*bp).y = rp.y;
170:  }
171:
172: void       calc_parm( bp,rp )
173: /* 移動用パラメーターの算出 */
174: b_parmPtr  bp;
175: r_parmPtr  rp;
```


▶「源平討魔伝」の義経のセリフは「殺してしんぜよう」だったと思います。
黒部　浩孝(21)三重県

```
176:
177:  {
178:     (*rp).x = (*bp).x;
179:     (*rp).y = (*bp).y;
180:     (*rp).drx = (*bp).drx;
181:     (*rp).dry = (*bp).dry;
182:     if( (*rp).dx > (*rp).dy )
183:             (*rp).f = (*rp).dx;
184:     else
185:             (*rp).f = (*rp).dy;
186:     (*rp).dxx = (*rp).dx;
187:     (*rp).dyy = (*rp).dy;
188:  }
189:
190: void        reflect( bp,rp,r )
191: /* 反射の処理 */
192: b_parmPtr   bp;
193: r_parmPtr   rp;
194: int         r;
195:
196:  {
197:  int         i,j,k,dx,dy,df,dxx,dyy;
198:     if( (*bp).drx == -1 )
199:             dx = -(*bp).dx;
200:     else
201:             dx = (*bp).dx;
202:     if( (*bp).dry == -1 )
203:             dy = -(*bp).dy;
204:     else
205:             dy = (*bp).dy;
206:     /* X方向の反射 */
207:     i = dx * co[r];
208:     j = dy * si[r];
209:     k = i + j;
210:     if( k > 0 )
211:     {
212:             (*bp).drx = 1;
213:             dxx = k >> 12;
214:             /* 三角関数のテーブルは4096倍して
215:                あるので、12ビット右にシフトする */
216:     }
217:     else
218:     {
219:             (*bp).drx = -1;
220:             dxx = (-k) >> 12;
221:     }
222:     /* Y方向の反射 */
223:     i = dx * si[r];
224:     j = dy * co[r];
225:     k = i - j;
226:     if( k > 0 )
227:     {
228:             (*bp).dry = 1;
229:             dyy = k >> 12;
230:     }
231:     else
232:     {
233:             (*bp).dry = -1;
234:             dyy = (-k) >> 12;
235:     }
236:     if( dx*(*bp).drx > 0 )
237:             (*bp).dx = dxx*13/16 + ((*bp).dx / 8);
238:     else
239:             (*bp).dx = dxx*13/16 - ((*bp).dx / 8);
240:     if( dy*(*bp).dry > 0 )
241:             (*bp).dy = dyy*13/16 + ((*bp).dy / 8);
242:     else
243:             (*bp).dy = dyy*6/8 - ((*bp).dy / 8);
244:     /* 反射後ボールがどれだけ動くかを計算 */
245:     dx = (*bp).dx >> 16;
246:     dy = (*bp).dy >> 16;
247:     if( (*rp).dxx > (*rp).dyy )
248:             df = (*rp).dx;
249:     else
250:             df = (*rp).dy;
251:     if( (*rp).f != 0 )
252:     {
253:             (*rp).dx = dx*((*rp).f-df)/(*rp).f;
254:             (*rp).dy = dy*((*rp).f-df)/(*rp).f;
255:     }
256:     else
257:     {
258:             (*rp).dx = 0;
259:             (*rp).dy = 0;
260:     }
261:     calc_parm( bp,rp );
262:  }
263:
264: void        locate_ball( bp )
265: /* ボールを最初に置く位置を決める */
266: b_parmPtr   bp;
267:
268:  {
269:  int         i = 0,j,k,x = 128,y = 128,mx,my;
270:     SP_ON();
271:     mouse( 4 );
272:     mouse( 2 );
273:     setmspos( 128,128 );
274:     while( i == 0 )
275:     {
276:             msstat( &j,&j,&i,&j );
277:             mspos( &mx,&my );
278:             if( point( mx,my ) != 4 )
279:             {
280:                     mx = x;
281:                     my = y;
282:                     setmspos( mx,my );
283:             }
284:             x = mx;
285:             y = my;
286:             put_ball( mx,my );
287:     }
288:     (*bp).x = mx;
289:     (*bp).y = my;
290:     while( i != 0 )
291:             msstat( &j,&j,&i,&j );
292:  }
293:
294: void        put_ball( x,y )
295: /* 画面にボールを表示する */
296: int         x,y;
297:
298:  {
299:     SP_REGST( 0,x+8,y+6,1<<8,3 );
300:     SP_REGST( 0x80000001,x+8,y+22,(1<<8)+1,3 );
301:  }
302:
303: void        draw_screen()
304: /* 画面を描く */
305:
306:  {
307:     screen( 0,1,1,1 );
308:     palet( 0,0 );
309:     palet( 1,65535 );
310:     palet( 2,rgb(  5,31,31 ) );
311:     palet( 3,rgb(  5, 8,31 ) );
312:     palet( 4,rgb(  5, 8,31 ) );
313:     fill( 0,0,255,255,2 );
314:     circle( 60,60,50,3,0,360,255 );
315:     paint( 60,60,3 );
316:     circle( 194,60,50,3,0,360,255 );
317:     paint( 194,60,3 );
318:     circle( 194,194,50,3,0,360,255 );
319:     paint( 194,194,3 );
320:     circle( 60,194,50,3,0,360,255 );
321:     paint( 60,194,3 );
322:     fill( 60,11,194,243,3 );
323:     fill( 10,60,244,194,3 );
324:     /* ボールを置ける範囲を4の色で描いておく */
325:     circle( 61,61,43,4,0,358,240 );
326:     paint( 60,60,4 );
327:     circle( 194,61,43,4,0,358,240 );
328:     paint( 194,60,4 );
329:     circle( 194,194,43,4,0,358,240 );
330:     paint( 194,194,4 );
331:     circle( 61,194,43,4,0,358,240 );
332:     paint( 60,194,4 );
333:     fill( 60,21,194,234,4 );
334:     fill( 18,60,237,194,4 );
335:  }
336:
337: void        create_map()
338: /* 接触情報マップを作成する */
339:
340:  {
341:  int         i,x,y;
342:  double      a,rr,p = 3.14159;
343:     for( x = 0; x != 256; x++ )
344:             for( y = 0; y != 256; y++ )
345:                     bmap[x][y] = 255;
346:     rr = (double)256/(double)238;
347:     /* 角の丸まった部分の接触情報 */
348:     for( i = 0; i != 91; i++ )
349:     {
350:             a = (double)i/(double)180;
351:             x = (int)(cos(a*p)*43+0.5);
352:             y = (int)(sin(a*p)/rr*43+0.5);
353:             bmap[60-x][60-y] =
354:                     (int)(33+(double)i/2.8125) % 64;
355:             bmap[195+x][60-y] =
356:                     (int)(32-(double)i/2.8125);
357:             bmap[60-x][195+y] =
358:                     (int)(32-(double)i/2.8125);
359:             bmap[195+x][195+y] =
360:                     (int)(33+(double)i/2.8125) % 64;
361:     }
362:     /* 平らな壁の接触情報 */
363:     for( i = 59; i != 196; i++ )
364:     {
365:             bmap[i][20] = 0;
366:             bmap[i][235] = 0;
367:             bmap[17][i] = 32;
368:             bmap[238][i] = 32;
369:     }
370:  }
371:
372: void        store_sctab()
373: /* 三角関数を4096倍して配列にストアする */
374:
375:  {
376:  int         i;
377:  double      a,p = 3.1415;
378:     for( i = 0; i != 64; i++ )
379:     {
380:             a = (double)i/(double)32;
381:             si[i] = sin( a*p )*(double)4096;
382:             co[i] = cos( a*p )*(double)4096;
383:     }
384:  }
385:
386: void        set_sprite()
387: /* スプライトとパレットの定義 */
388:
389:  {
390:  int         i;
391:     SP_INIT();
392:     SP_DEFCG( 0,1,&ballpat[0][0] );
393:     SP_DEFCG( 1,1,&ballpat[1][0] );
394:     for( i = 0; i != 16; i++ )
395:             SPALET( i,1,pal[i] );
396:  }
```

▶私は必ず有名人になる。フフフ……。　　帆足　徹也(18)福岡県

Oh!X

LIVE in '94

X68000・Z-MUSIC用
(SC-55対応)

「宇宙戦艦ヤマト」より

THE BIRTH・誕生

Hayasaka Makoto
早坂 真

X68000・Z-MUSIC+
PCM8用

「プロジェクトA子」より

SPACESHIP IN THE DARK

Watanabe Kazuhiko
渡辺 一彦

今月は，Oh!X読者のうちの大部分を占めるというウワサも囁かれているアニメファンのみなさまに捧げるLIVE inです。誰でも知ってる曲とマニア向けの1品という絶妙(?)の選曲。アニメにうとい人でも聴いてソンなしのお勧めの2作品です。

波動砲・発射っ！

さて今月はアニメで攻めてみたいと思います。1曲目は誰もが知っている作品からいってみましょう。名作「さらば宇宙戦艦ヤマト」より，「宇宙戦艦ヤマト～THE BIRTH・誕生～」をお届けしましょう。SC-55同等品が必要になります。

この作品の完成度はハンパではありません。投稿されたその日のうちに採用決定，あとは掲載号を決めるだけ……しかし，あまりのリストの長さに，やはりボツか……，でもこの完成度の高さは……と検討の末，ここに復活を果たすという，まさにヤマトのようなドラマがあったのでした。

そのドラマにふさわしいこの作品，有名な「あの」ヤマトがここに再現されると思えば間違いないでしょう。作者の早坂さんはパソコン通信で多くの人に聴いてもらい，さらに自分でも何度も聴き直して作品を完成させています。マーケティング戦略が効を奏しているのでしょう。参考にしたCDが「交響組曲・宇宙戦艦ヤマト」というだけあって，シンフォニックなサウンドには感動すら覚えます。全部で4分40秒にわたる超大作，惜しみない拍手の準備をどうぞ。

ちょっと長いリストですが，入力したところまで聴ける構成になっているので，がんばってください。きっと最初のあたりを聴いただけで，「タダモノではないっ！」と思うはず。あとはイッキに最後まで入力してしまいましょう。今回は特別にカウンターを数回に分けて掲載しますので，入力ミスのチェックも大丈夫ですよね？

販促プログラム for ver.2.0

さて，次に紹介するのは「プロジェクトA子」というOVA(オリジナルビデオアニメ)です。1986年頃の作品なので，かれこれ8年近くも前になりますか。詳しく知っている人は少ないかもしれませんね。曲はサントラ盤に収録されている「SPACESHIP IN THE DARK」です。演奏にはPCM8.Xが必要ですが，内蔵音源だけですので，MIDIを持っていない人でも大丈夫です。

作品はよくポイントを押さえています。限られたPCMデータのなかでよくやっているといえるでしょう。強いていえば，ディストーションギターがちょっと淋しいかな。ラストが突然のフェードアウトなので，ちょっと違和感がありますが，これは元曲でもそうなっているので「そこまで再現している」と考えてくださいね。

サンプリングデータはOh!X BOOKS「Z-MUSICシステムver.2.0」のディスク4に収録されたデータを加工して使用しています。Z-MUSIC ver.2.0がないと正しい演奏にならないので，ちゃんと手に入れてくださいね。加工の仕方は以下のとおりです。

1) ZVT.Xを起動して，disk4 ¥EXTRA_PERC¥J_BELL.PCM をロードする
2) 「E」キーを押して，8を選択
3) START POINTを200にセット
4) 「PITCH」を選択し，12halfトーン(1オクターブ)上げる
5) JBELL1.PCM のファイル名でセーブ

あとは，SPACE.CNFを入力，ZPDファイルを作成すればOKです。

渡辺くんは投稿するにあたり，オリジナルのサンプリングデータから汎用(?)のサンプリングデータへとコンバートしたそうです。やっぱり作ったからには多くの人に聴いてもらいたいもんね。

担当者より

最近の投稿では，桁数をまったく無視しているものが目立ちますが，掲載されたときの美しさやわかりやすさを考えたリストのほうが，より多くの人に聴いてもらえるのではないでしょうか。掲載時には64桁で印刷していますので，参考にしてください。

あと，作成時の苦労話や，作るきっかけ，元曲に対する思い入れ，LIVE in への意見，そのほか世間話など，なんでもいいからドキュメントにして同封していただけると嬉しいなぁ。 (SIVA)

宇宙戦艦ヤマト

プロジェクトA子

リスト1　宇宙戦艦ヤマト〜THE BIRTH・誕生〜

日本音楽著作権協会(出)許諾第9373076-301号

```
  1: / YAMA_BIRTH.ZMS  1994-01-10
  2: .comment 宇宙戦艦ヤマト〜THE BIRTH・誕生〜 for SC-55 by Yasaka
  3:
  4: /（T1）と（T12）は、他のトラックに同期信号を送っている
  5: /ので、打ち込みの際は音長に注意して下さい。
  6:
  7: (i)
  8: (m1,5000)(aMidi01,1)   / ｽﾄﾘﾝｸﾞｽ 1
  9: (m2,5000)(aMidi01,2)   /        2
 10: (m3,5000)(aMidi02,3)   /        3
 11: (m4,5000)(aMidi02,4)   /        4
 12: (m5,5000)(aMidi03,5)   / ﾃｨﾝﾊﾟﾆｰ
 13: (m6,5000)(aMidi04,6)   / Brass 1
 14: (m7,5000)(aMidi05,7)   /       2
 15: (m8,5000)(aMidi06,8)   /       3
 16: (m9,5000)(aMidi07,9)   / E.ﾍﾞｰｽ
 17: (m10,5000)(aMidi08,10) /
 18: (m11,5000)(aMidi09,11) / ｽｷｬｯﾄ
 19: (m12,5000)(aMidi10,12) / Drum 1
 20: (m13,5000)(aMidi10,13) /      2
 21: (m14,5000)(aMidi11,14) / C_SD
 22: (m15,5000)(aMidi11,15) / ﾀﾝﾊﾞﾘﾝ
 23: (m16,5000)(aMidi12,16) /
 24: (m17,5000)(aMidi13,17) / Brass
 25:
 26: .roland_exclusive 16,$42 ={$40,$00,$7F,$00}
 27: .sc55_reverb $10={03,03,00,100,74,100,16}
 28: .sc55_chorus $10={02,02,64,08,80,03,19,00}
 29: .sc55_v_reserve $10={2,2,1,1,2,3,1,1,1,3,3,2,2,0,0,0}
 30: .sc55_PART_SETUP 14,={17}
 31: .sc55_PART_SETUP 15,={17}
 32: .sc55_PART_SETUP 16,={17}
 33: .sc55_Print $10="YAMATO - THE BIRTH -"
 34:
 35: /B0
 36: (t1) @I$41,$10,$42  @u90  @v85  @C7 @G12
 37: (t2) @I$41,$10,$42  @u80
 38: (t3) @I$41,$10,$42  @u80  @v85  @C7 @G12
 39: (t4) @I$41,$10,$42  @u90
 40: (t5) @I$41,$10,$42  @u100 @v127 @C7 @G12
 41: (t6) @I$41,$10,$42  @u50  @v70  @C7 @G12
 42: (t7) @I$41,$10,$42  @u80  @v70  @C7 @G12
 43: (t8) @I$41,$10,$42  @u100 @v100 @C7 @G12
 44: (t9) @I$41,$10,$42  @u67  @v100 @C7 @G12
 45: (t10) @I$41,$10,$42 @u60  @v100 @C7 @G12
 46: (t11) @I$41,$10,$42 @u50  @v100 @C7 @G12
 47: (t12) @I$41,$10,$42 @u127 @v110     @G12
 48: (t13) @I$41,$10,$42
 49: (t14) @I$41,$10,$42 @u127 @v110     @G12
 50: (t15) @I$41,$10,$42
 51: (t16) @I$41,$10,$42 @u80  @v60  @C7     @G24
 52: (t17) @I$41,$10,$42 @u60  @v75  @C7 @G12
 53:
 54: /B1
 55: (t1)  r1 r8@50 @p25  @B0 @Y1,99,54r4 @E60,70 r4t63
 56: (t2)  r1 r8 r2
 57: (t3)  r1 @50r8 @p100 @B0 @Y1,99,54r4 @E60,70 r4
 58: (t4)  r1 r8 r2
 59: (t5)  r1 r8@48 @p64  @B0 r4 @E72,40 r4
 60: (t6)  r1 @45r8 @p30  @B0 r4 @E65,60 r4
 61: (t7)  r1 r8@45 @p75  @B0 r4 @E65,60 r4
 62: (t8)  r1 @58r8 @p64  @B0 r4 @E70,30 r4
 63: (t9)  r1 r8@35 @p64  @B0 r4 @E50,10 r4
 64: (t10) r1 @73r8 @p64  @B0 r4 @E70,30 r4
 65: (t11) r1 r8@69 @p64  @B0 r4 @E80,38 r4
 66: (t12) r1 r16   @p64  @B0 @E127,0 x$40,$10,$15,$01 @1 r16
 67: /     Volume         Panpot       Reverb depth   Chorus depth
 68: (t12) r32@Y26,35,100 @Y28,35,64  r32@Y29,35,15  @Y30,35,00
 69: (t12) r32@Y26,40,80  @Y28,40,64  r32@Y29,40,65  @Y30,40,40
 70: (t12) r32@Y26,42,90  @Y28,42,80  r32@Y29,42,35  @Y30,42,00
 71: (t12) r32@Y26,46,80  @Y28,46,74  r32@Y29,46,35  @Y30,46,00
 72: (t12) r32@Y26,57,100 @Y28,57,55  r32@Y29,57,40  @Y30,57,00
 73: (t12) r32@Y26,53,70  @Y28,53,55  r32@Y29,53,40  @Y30,53,00
 74: (t12) r32@Y26,48,80  @Y28,48,64  r32@Y29,48,20  @Y30,48,00
 75: (t12) r64@Y26,45,80  @Y28,45,64  r64@Y29,45,20  @Y30,45,00
 76: (t12) r64@Y26,43,80  @Y28,43,64  r64@Y29,43,20  @Y30,43,00
 77: (t13) r1 r8 r2
 78: (t14) r1 r8 x$40,$1A,$15,$02 r16 @49 r16 @B0 @p64 @E127,0
 79: /     Volume         Panpot       Reverb depth   Chorus depth
 80: (t14) r32@Y26,57,85  @Y28,57,40  r32@Y29,57,80  @Y30,57,00
 81: (t14) r32@Y26,38,127 @Y28,38,64  r32@Y29,38,30  @Y30,38,40
 82: (t14) r32@Y26,29,95  @Y28,29,64  r32@Y29,29,75  @Y30,29,00
 83: (t14) r32@Y26,54,70  @Y28,54,64  r32@Y29,54,35  @Y30,54,00 r8
 84: (t14) @Y24,38,61
 85: (t15) r1 r8 r2
 86: (t16) r1 r8@2  @p64  @B0 r4 @E30,20 r4
 87: (t17) r1 r8@41 @p25  @B0 r4 @Y1,99,72 @E40,20 r4
 88:
 89: /B2        A－1（イントロ）
 90: (t1)  @q1o3g2<f2& _5|:3f16&_5:|f*9~20r*3t5918efect63d4&
 91:       d*45r*3t59c+dc+>a<t63c4& c*45r*3>t60g+b-g+2
 92: (t3)  @q1o2g2<f2& _5|:3f16&_5:|f*9r*3~2018efecd4&
 93:       d*45r*3c+dc+>a<c4& c*45r*3>g+b-g+2
 94: (t9)  x$40,$17,$32,$13,$0E,$30,$54,$40 r1 r1 r1 r1
 95: (t12) r1 r1 r1 r1
 96: (t17) @q1o3g2<f2& _5|:3f16&_5:|f*9~20r*318efecd4&
 97:       d*45r*3c+dc+>a<c4& c*45r*3>g+b-g+2
 98:
 99: /B3        A－2
100: (t1)  w16116|:12g&_2:| |:16g&_1:| t6018|:8g&_2:| |:7g&_1:|g
101:       r4@v85t46@q2ggt65
102: (t3)  116|:12g&_2:| |:16g&_1:| 18|:8g&_2:|
103:       |:7g&_1:|g r4@v85@q2gg
104: (t9)  o1q8g4.r8g4.|:3u-4r8g4.r8g4.:| r4.
105: (t12) w13r1 r1 r1 r1 r4
106: (t13) w14@u34o3|:4rfrf:| r
107: (t16) w14ro6'ab-',1r<'de-' r'c+d',0r>u-7'ab-'
108:       _8ru-7'dff+'ru-7'dff+',0 ru-7'dff+'ru-7'dff+' r
109: (t17) 116|:12g&_2:| |:16g&_1:| 18|:8g&_2:|
110:       |:7g&_1:|g r4@v75@q2gg@q1
111:
112: /B4        A－3
113: (t1)  w11@q1<c2^8t60cd>b- t64a&|:9a16&_4:|a32r32t60@q2
114:       ~15g~15g @q1<~10t67c2^8t62e-dc t50|:15f+16&_6:|f+16
115: (t3)  @q1<c2^8cd>b- a&|:9a16&_4:|a32r32@q2~15g~15g
116:       @q1<~10c2^8e-dc |:15f+16&_6:|f+16
117: (t11) w@Y1,99,66r1 r1 r1 r*190
118: (t12) r1 r1 r1 r1
119: (t17) <c2^8cd>b- a&|:9a16&_4:|a32r32@q2~15g~15g
120:       @q1<~10c2^8e-dc |:15f+16&_6:|f+16
121:
122: /B5        A－4
123: (t1)  w6w7r1 w10r1 @Y1,99,64r1 r1 r@u80@v40o3'dg+<dg+'&
124: (t3)  @Y1,99,64r1 r1 r1 r1 @49@u80@v35ro2d&
125: (t6)  wo4@A40,50,60,70,60,40,20'df'*336@A30,35,40,35'e-4f+'
126:       u-8@A40,50,65,75,65,55,45.30'eg+'*384 r4
127: (t7)  wo3@u50@A40,50,60,70,60,40,20g+*384 b-*384 r4@A
128: (t10) r1 wr1 t60o518cc+t58g4.t61f+e-t60d
129:       cc+t58g4&|:11g16&_6:|g16
130: (t11) t60o418u-10b<u+5ct59@H24@M64uf+4.@Mt60fdc+
131:       >b<ct56f+4&|:7f+16&_6:|f+16 r*2 r1 r1 r4
132: (t12) r1 r1 r1 r1 r4
133:
134: /B6        A－5
135: (t1)  w8@u70'dg+<dg+'*360 @D1'c+1g<c+g'|:13r8_4:|@D0
136: (t3)  d*360 @D1c+1|:13r8_4:|@D0
137: (t6)  r1 r1 r1 @Ar2 I8@62@u110@v70@p105@E60,30r1
138: (t7)  @p30@u77@v80o518t65u-15fcu+10bt60ub-4.t65u-15fc
139:       u+10bt61ub-&|:7b-16&_4:|b-16r4
140:       ~28t65u-15e>b<u+8b-t62ua4.u-16e>b
141:       u+8<b-t57ua4.&|:15a16&_6:|a16
142: (t8)  wr1 r1 r1 r1 r2o3116q3
143: (t12) w14r1 r1 r1 r2 r1
144: (t14) wr1 r1 r1 r2 r2r4..t88q4Z22,26,30o2{ddd}16
145:
146: /B7        A－6
147: (t1)  w9w17t88r1 r1 t89r1 @49@u90@v55@p18r1t88
148: (t3)  r2@58r2 @Y1,99,55@u110@v90@E60,30@p25r1o3
149:       q3b4b.q4116b<q8@A90,,80,75,70,65,60,55
150:       e8.r@A@v90q4g+f+er f+8.>bq8|:4b&_9:||:7b&~3:|b
151: (t6)  r1 q3o4116e4e8.q4eq8a4<@Y1,99,66q5c+16>bar
152:       b8.<@Y1,99,64eq8e4&|:8e16&~2:| |:15e16&~2:|e16
153: (t8)  a4a8.q4a<@A100,,90,85,80,75,70,60q8d8.rq4@A@v100f+edr
154:       e8.a16q8a4&|:8a16&_2:| a1& a1
155: (t9)  wr1 @u117@v80@p64r1 r1 r1
156: (t12) r1 r1 r1 r1
157: (t14) |:@u40d4r8.Z22,26,30{ddd}16@u40d4r8.Z22,26,30{ddd}16
158:       |:@u40d8.Z22,26,30{ddd}16:||@u40d4r8.Z22,26,30{ddd}16
159:       :| @u40d8@u30116dddddd
160: (t17) w@Y1,99,64r1 r1 r1 @57@E30,50@v55@p100r1
161:
162: /B8        A－7
163: (t1)  w7w11w16r1 @Y1,99,60 @E60,80r1 t87r1 r1
164: (t3)  @Y1,99,64r1 r2@49@u90@v55@p18@Y1,99,60@E60,65r2 r1 r1
165: (t6)  r1 r1 10@62@u100@v70@p95@E70,40r1 r1116
166: (t7)  wr1 r1 r2@62@u110@v70@p95@E70,40r2 r2@Y1,100,110r2
167: (t8)  @Y1,32,78r1 r1 r1 @u120@v65@p35@E30,35r*190
168: (t9)  r1 r1 r1 @E40,10@Y1,$66,$2Er1
169: (t11) w@Y1,99,64r1 @45@u100@v50@p70@E70,40 r1 r1 r1
170: (t12) r1 r1 r1 r1
171: (t14) |:4|:@u33d8Z10,10,15{ddd}16Z15,20{dd}16:| @u33d8
172:       Z5,10,10{ddd}16Z15,15,20{ddd}16 Z20,25,30,30{dddd}4:|
173: (t16) w@77@u100@v95@p70q8@B0@E100,20r1 r1 r1 r1
174: (t17) @Y1,100,110r1 r1 r1 r1 116
175:
176: /B9        B－1（ここから盛り上がります！）
177: (t1)  t158r1 o4116q4f8.fq3f8f8q6fb-<r4.
178:       >q4b-8.b-q3b-8b-8q6b-<e-r4. q4e-8.e-q3e-8e-8q6e-a-r4.
179: (t3)  r1 o4@D1r2r8<q8c4. r2r8f4. r2r8b-4.
180: (t6)  r1 o4q5f8.q4fq3f8f8q5u+10fb-r4.
181:       u-5b-8.q4b-q3u+5b-8b-8q5u+7b-<e-r4.
182:       u-5e-8.q4e-q3u+5e-8e-8q5u+7e-a-r4.
183: (t7)  r1 o4q8@D1r2r8<c4. r2r8u+7f4. r2r8u+7b-4.
184: (t8)  o3116q4c8.cc8c8q5cfq8g4.
185:       q4f8.fq3f8f8q5fb-<q8c4.
186:       >q4b-8.b-q3b-8b-8q5b-<e-q8f4.
187:        q4e-8.e-q3e-8e-8q5e-a-q8@D1b-4.&
188: (t9)  o2116q4c8.cc8c8q6c*21q8f*3g4.& |:16g8&_2:| g1&
189: (t11) r2^8@D1o4g4. r1 r2^8f4. r1
190: (t12) r1 r1 r1 r1
191: (t14) o1q4@u65f16 Z50,55{ffff ffff ffff ffff f}*180
192:       |:3{ffff ffff ffff ffff f}1:|
193: (t17) @u95o4q5c8.cq3u+7c8c8q4u+7_5cfq8_5u+7@D1g4. r1 r1 r1
194:
195: /B10       B－2
196: (t1)  w5r1 r1 r1
197: (t3)  r2|:8r16~1:| r4|:7r8~4:|~8r8@D0 r2.
198: (t5)  w@v85q2o3@u115c16Z65,70,67,62{cccc cccc cccc cccc}*180
199:       |:{cccc cccc cccc cccc c}1:|
200: (t6)  r1 r1 r1
201: (t7)  r2|:8r16~2:| r4|:15r16~2:|~8r16@D0 r2.
202: (t8)  r2&|:8r16&~1:| r4&|:6r8&~4:|r8&~8r8 @D0r2.@Y1,32,64r*2
203: (t9)  g1& g1 r1
204: (t11) r2|:8r16~1:| r4|:15r16~2:|~8r16@D0 r2.
205: (t12) r1 r1 r1
206: (t14) |:3{ffff ffff ffff fffff}1:|
207: (t17) r2&|:8r&~2:| r4&|:15r&~2:|r @D0r2.
208:
209: /B11       B－3
210: (t1)  w10r1 @u110@v90@p18@E80,80r2
211: (t5)  {cccc cccc ~3cc~5cc ~ccc~cc}1 {c~c~c~c}4r4@v127@E50,40
212: (t6)  r1 @57@u110@v90@p100@E70,20@Y1,99,64r2
213: (t7)  @Y1,99,60r1 @Y1,100,64r4@u100@v90@p95@E70,20r4
214: (t8)  @58r1 r4@v100@p35@E65,35@A100,,95,90,85,80,75,70r*46
215: (t9)  r1 r4@u117@v90r4
216: (t10) wr2@74@u100@v80@p80@B0@E30,30r2 r2
217: (t11) r1 r2
218: (t12) r1 r2
```

```
219: (t14) {ffff ffff ~3ff~ff ~fff~ff}1
220:       Z50,56,63,70{f~ff~f}4r16@Y29,38,50r16@Y26,38,110r8
221: (t17) @61@v95 @p70r2@E40,20@Y1,100,64@B0r2 r2
222:
223: /B12       B－4（ここを聴けばヤマトの曲だとわかりますね）
224: (t1)  r1 r1 r1 w2r2.o614c+
225: (t2)  w@u60@p18r2.o514a
226: (t5)  o214@u80fcfc fcfc fcfc fcfu-20c
227: (t6)  o4q6t122c4t14318c.u-8c16uq8t145f4q4af
228:       g.<q5116u-5cq8uc4&|:7c32&_7:|c32~49q5u-8dc>b-r
229:       u<q418c>fa<cq8e4q5u-8116fefr
230:       uq8c4&|:7c32&_6:|c32r8@E50,r8@u90@v80t13514c+
231: (t7)  o4q6c418c.u-8c16uq8f4q4af
232:       g.<q5116u-5cq8uc4&|:7c32&_7:|c32~49q5u-8dc>b-r
233:       u<q418c>fa<cq8e4q5u-8116fefr
234:       uq8c4&|:7c32&_6:|c32r8@E40,r8@u110@v90>14'a<c+'
235: (t8)  @u115o3116r2q4c4c8.q6@A100c @A100,,95,90,85,80,75,70
236:       q4f4a4g8.<@A100q6u-8cu@A100,,95,90,85,80,75,70c4
237:       @A100r4u-8dc>b-r<q4uc8>f8a8<c8
238:       u+8@A100,,95,90,85,80,75,70e4q6@A100
239:       ufefru-8q7cr@A@E40,r*26>>@v90q814'a<a<c+'
240: (t9)  o2q614fcfc fcfc fcfc fcfa
241: (t10) r1 r1 r1 r2.o6q6@u60a4
242: (t12) w13q3@u50o114bbbb bbbb bbbb bbb@u65b
243: (t13) wq2r1 r1 r1 r2.o3@u8014a
244: (t14) @v110q3o2116|:4@u50d8@u40dd@u50dZ35,30ddd
245:       @u50d8@u40dd@u50dZ35,30ddd:|
246: (t17) @u60o418q6c4c.c16q8f4q4af
247:       @u70g.<@u110116q5cq8c4&|:7c32&_5:|c32~35q6@u100dc>b-r
248:       18<q4c>fa<cq8e4q6@u50116fefr
249:       @u10014q8c&|:7c32&_5:|c32r~35@u58q6c+
250:
251: /B13       B－5
252: (t1)  q6u+5dq4uc8>b-8<q6c>u-6a t+1ub-q4a8g8q6au-7f
253: (t2)  q6u+5b-q4ua8g8q6au-6f @u75eq4e8e8q6fu-7c
254: (t5)  r1 r1
255: (t6)  q6u+6dq4uc8>b-8<q6c>u-5a ub-q4u+4a8g8uq6au-5f
256: (t7)  q6u+6'b-<d'q4u'a8<c''g8b-'q6'a<c'u-5'fa'
257:       u'eb-'q4'e8a''e8g'q6u'fa'u-5'df'
258: (t8)  q6u+5'b-<b-<d'q4'a8<a<c''g8<gb-'q6u'a<a<c''f<fa'
259:       u+5'g<gb-'q4u'e8<ea''e8<eg'q6'f<fa'u-4'd<df'
260: (t9)  b-4a8g8a4f4 g4f8e8f4d4
261: (t10) a4q4a8g8q6a4f4 e4q4e8e8q6f4f4
262: (t12) brbb brbb
263: (t13) |:u-10aru-10aua:|
264: (t14) r1 r1
265: (t17) q6dq4c8>b-8<q6c>a b-q4a8g8q6af
266:
267: /B14       B－6
268: (t1)  w3t139@45q8@u90_13o5g1 t+1<c1
269: (t2)  q8@u75o4g1 <c1
270: (t3)  wr1 @Y1,$21,100 @Y1,$20,50 r2@31@v35r4@E65,20@p74r4
271: (t5)  r1 @8@E65,30@v62@p80@Y1,$20,54 @Y1,$21,84r1
272: (t6)  @E50,@u100@v95116q5|:3o4u-8cufg|r:|o4q7c8q6fgr
273:       |:3o4u-8gub-<c|r:|o4q7g8q6b-<cr
274: (t7)  @E50,@u110@v95116q6|:3o4u-8cufg|r:|o4q7c8q6fgr
275:       |:3o4u-8gub-<c|r:|o4q7g8q6b-<cr
276: (t8)  @u110@v100116q5|:3o3cfg|r:|o3q7c8q5fgr
277:       |:3o3gb-<c|r:|o3q7g8q6b-<cr
278: (t9)  |:c4c4c8.c8c16c16r16:|
279: (t12) |:b4b4b8.b4r16:|@Y29,40,45
280: (t13) r1 r1
281: (t14) @u50|:|:dddr:|dddd8ddr:|
282: (t17) @u110@v90o4q2'cg''cg''c8.g''c8g'116'cg''cg'r
283:       'g4<c''g4<c''g8.<c''g8<c''g<c''g<c'r
284:
285: /B15       C－1（T3とT5は・・・変です ^^;）
286: (t1)  t143116@E50,@49@u110@v95o4q7c4q6c8.u-15cuf4q5f8f8q4
287:       g8.<q6u-15cq8ut+1c4.&_4|:7c&_3:|cr8
288:       t-1~25q6dc>b-r<q8c8frq7fefr
289:       t+1q8@B-1366,0,0c&c8&_3|:6c&_3:|cr8~21@B0q7c8f8
290: (t3)  116r2.o4@u70q4'cfg'@u115_5q6'cfg'@u90~5q5'cfg'r
291:       @v45|:3r2.o4@u70q4'cfg'@u115_5q6'cfg'@u90~5q5'cfg'r:|
292: (t5)  o418@u90q2cru-10'fb-'r'fb-'r'cg'r ur4.cru-10'fb-'r4
293:       'fb-'r'cg'u'cg'ru-8'fb-'r4 r4.'cg'r'fb-'r'cg'
294: (t9)  116@v80@q2o2|:c8>b-b-b-8b-8<cc8c>b-b-8b-
295:       <cc8ddd8e-e-e-8e-f8g8:|
296: (t12) w1514|:3@u75b<e>b<e>:| 18b<e16e.>b<{eeeeeeee}2
297: (t13) 18o2<@u80a*0>|:4@u75f+f+|@u90b-4:|@u90b-b-
298:       |:@u75f+f+|@u90b-4:|@u90b-b- f+r4r@u100b-4r4
299: (t14) @Y29,38,30 @u6514|:3rdrd:| r8d16d8.r8{dddddddd}2
300: (t15) wZ50,50o3116|:4|:4f+f+f+f+:|:|
301:
302: /B16       C－2
303: (t1)  t-1q8g8q7ab-<cr>fr<dc>b-rq4t+118aq6f
304:       q3gcafq8116u+5efudrq618de q3c>b<d>gq7116gafr<q4g8f8
305:       t-1q8u+5f8uq7edq418ecq3dgdb
306: (t3)  |:4r2.o4@u70q4'cfg'@u115_5q6'cfg'@u90~5q5'cfg'r:|
307: (t5)  u-20ccru+10c'fb-'ucr4 r2u-10cu'fb-'r
308:       'fb-' |:u-15r2cc|r4:|r'fb-'
309: (t9)  c8>b-b-b-8b-8<c16c16c+8c8>b-8 <cc8c>b-8b-8<cc>b-8<e-8ff
310:       ggg8g.r32ggggg8fe->b-8 <c8ccf8g8ggg8ggg8
311: (t12) 14|:3>b<e>b<e>:| b16<e8.@u65<{cc>ar}4
312:       @u75e16e8.{ee@u65<c>a}4
313: (t13) <@u80a*0>@u75f+f+@u90b-4@u75f+f+@u90b-b-
314:       @u75f+f+@u90b-b-@u75f+f+@u90b-4 |:@u75f+f+@u90b-4:|
315:       @u110b-4r4@u110b-4r4
316: (t14) |:3rdrd:| r16d8.rd16d8.{ddrr}4
317: (t15) |:4|:4f+f+f+f+:|:|
318:
319: /B17       C－3
320: (t1)  w6w7u+8q8<c>q7ub16a16q4bgq3u+5t+1a<d>a<c
321:       t-1w8c>bbaaq4guq8g<f
322:       e116dq6u+8c8uq8c4.r>q7gfed e8.q8dc2.
323: (t3)  |:4r2.o4@u70q4'cfg'@u115_5q6'cfg'@u90~5q5'cfg'r:|
324: (t5)  |:ucu-20c>c<b-u-5rc>cr< u+10cu-10fr>c<crb-r:|
325: (t6)  w@v90@p100@E70,20r1 r1 r1 r1
326: (t7)  wr1 @v90@p95@E60,20r1 r1 r1
327: (t8)  r1 wr1 @v100@p35@E55,35@A100,,95,90,85,80,75,70r1 r*190
328: (t9)  g8ddd8dddfg8<@B0,-4641,24d8.>@B0g g8ffe8efga<c8e8f8
329:       cc8>b-b-b-8aaa8ggffd cc8cdd8df8ffgg8g
330: (t12) >|:3@u75b<e>b<e>:| 'b16<e'<e8. 116ee@u65<ce>aaaa<ce>aa
331: (t13) <@u80a*0>|:6@u75f+f+|@u90b-4:|@u90b-b- @u70|:f+f+f+f+:|
332: (t14) |:3rdrd:| 116dd8.ddu-20dd |:dddd:|u
333: (t15) |:4|:4f+f+f+f+:|:|
334:
335: /B18       C－4
336: (t1)  w17t136r1 r1 r1 r1
337: (t6)  @u100o4q6d418d.d16q8g4q4bg
338:       a.<q5116u-5dq8ud4&|:7d32&_7:|d32~49q5u-8edcr
339:       uq418d>gb<dq8f+4q5u-8116gf+gr uq8d4&|:7d32&_6:|d32r2
340: (t7)  @u100o4q6d418d.d16q8g4q4bg
341:       a.<q5116u-5dq8ud4&|:7d32&_7:|d32~49q5u-8edcr
342:       uq418d>gb<dq8f+4q5u-8116gf+gr uq8d4&|:7d32&_6:|d32r2
343: (t8)  K2@u115o3r2q414cc8.q6@A100c16 @A100,,95,90,85,80,75,70
344:       q4fag8.<@A100q6u-5c16u+8@A100,,95,90,85,80,75,70c
345:       @A100ur116dc>b-r<q418c>fa<c u+4@A100,,95,90,85,80,75,70
346:       e4q6u@A100116fefru+4q7cr*14@Ar4.@v100
347: (t9)  o2@v90q614gdgd gdgd gdgd gdgr
348: (t12) 14q3@u50o114bbbb bbbb bbbb bbr@u100o3{c>agr}4
349: (t13) <@u80a4>r2. r1 r1 r1
350: (t14) q3o2@v120|:4@u60d8@u50dd@u60dZ45,40ddd@u60d8|@u50dd
351:       @u60dZ45,40ddd:|@v110r4.
352: (t17) wK2@u60@v95@p70o418q6c4c.c16q8f4q4af
353:       @u70g.<@u110q5116cq8c4&|:7c32&_5:|c32~35q6@u100dc>b-r
354:       <q418c>fa<cq8e4q6@u50116fefr @u100q8c4&|:7c32&_5:|c32r2
355:
356: /B19       C－5
357: (t1)  w3w5K2t143116@u110@v95o4q7c4q6c8.u-15cuf4q5f8f8q4
358:       g8.<q6u-15cq8ut+1c4.&_4|:7c&_3:|cr8
359:       t-1~25q6dc>b-r<q8c8frt+1q7fefr
360:       q8@B-1366,0,0c&c8&_3|:6c&_:|cr8~21@B0q7c8f8
361: (t3)  w|:4r4o4@u70q4'cfg'@u115_5q6'cfg'@u90~5q5'cfg'rr2:|
362: (t5)  w|:u-25ddgdrddg d16u+8d16u-8'dg'rd'ea'rdur:|
363: (t9)  K0116@v80@q2d8.df+f+f+8aaa8<d8>a8 dd8.aa8d<def+8g8d8
364:       >d8d8ef+g8aaa8<c+8d8 dd8dd8d8d>af+8e8d8
365: (t12) w15@u75o1|:3b<e>b<e>:| b<eu-15{eeu+5eu+5eu+5eeee}2
366: (t13) @u80<a*0>|:4@u75f+f+@u90b-@u70f+:|
367:       @u75f+f+@u90b-@u70f+@u75f+f+@u90b-@u75b-
368:       f+f+@u90b-b-b-4@u70f+4
369: (t14) @u6514|:3rdrd:| rdu-15{ddu+5du+5du+5dddd}2
370: (t15) w|:4|:4f+f+f+f+:|:|
371:
372: /B20       C－6
373: (t1)  t-1q8g8q7ab-<cr>fr<dc>b-rq418aq6f
374:       q3gcafq8116u+5efudrq618de q3c>b<d>gq7t+1116gafr<q4g8f8
375:       t-1q8u+5f8uq7edq418ecq3t+1dgdb
376: (t3)  |:4r4o4@u70q4'cfg'@u115_5q6'cfg'@u90~5q5'cfg'rr2:|
377: (t5)  u-20drd16d16adrrd 'df''fa'r4dded
378:       r2'c>g''c>g''d>a''d>a' u+8'd4>a''da'u-8rf+drru
379: (t9)  d8.deeffg8g8aa8a >a<a8abb8<cc8c+c+d8d8
380:       >>a8a8g+ad<d>g+a<d8ab<c+>d <d8d8ddd8ddd8dd>ad
381: (t12) >@u75|:3b<e>b<e>:| @u65bb16<e8.>b16<e8.>{b<e<c>a}4
382: (t13) <@u80a*0>|:|:@u75f+f+|@u90b-4:|@u90b-b-:|
383:       |:@u75f+f+@u90b-4:|@u75f+4@u120b-.b-.b-.r.
384: (t14) @u65|:3rdrd:| @u60rr16d8.r16d8.r16d8.
385: (t15) |:4|:4f+f+f+f+:|:|
386:
387: /B21       C－7
388: (t1)  t-1w6w7u+8q8<c>q7ub16a16q4bgq3u+5t+1a<d>a<c
389:       t-1w8c>bbaaq4guq8g<f
390:       e116dq6u+8c8uq8c4.r>q7gfed e8.q8dc2.
391: (t3)  |:4r4o4@u70q4'cfg'@u115_5q6'cfg'@u90~5q5'cfg'r|r2:|
392:       r4@49@u100@v75@p120@E30,65r8@Y1,$21,64@Y1,$20,64r8
393: (t5)  u-20'd.f+''c16e'{'eg''ce''ce'r}4r2
394:       u+5'e.g''d16f+'{'eg''df+''ce'r}4r2 u-5d.d16r4dddd
395:       d16d8d16f+r8r4dd @48@E50,40@Y1,$20,64 @Y1,$21,64
396: (t6)  wr1 @u100@v90@p100@E70,20@Y1,32,64r1 r1 r1
397: (t7)  wr1 r1@u100@v90@p95@E70,20r1 r1
398: (t8)  r1 wr1 r1@p35@E65,35@A100,,95,90,85,80,75,70 r*190
399: (t9)  >a8gggg+aga8ggga<c+8 >a8aga8a8aaa8a<c+>a8
400:       <d8d8f+f+f+8ggg8a8a8 <ddrd>aa<d>aa<c+d>aa<d>a8
401: (t12) >@u75|:3b<e>b<e>:| bu-20116<eue8.eeeeee@u60<c>a
402: (t13) <@u80a*0>|:@u90rb-rb-:| |:@u75f+f+|@u90b-4:|@u90f+b-
403:       |:@u75f+f+@u90b-@u65b-:| @u110b-4r4b-4r4
404: (t14) |:3rdrd:| r116dd8.dddddddd@Y1,29,50
405: (t15) |:4|:4f+f+f+f+:|:|
406:
407: /B22       D－1（だいたいリストの中間地点です。）
408: (t1)  w10w17K0t135r1 r1 t+1r1 w2@v90r2.o6t-114c+
409: (t2)  wr2.@u60o514a
410: (t5)  o214@u80@v100@p64fcfc fcfc fcfc fcfu-10c
411: (t6)  K0o4q6c418c.u-6c16q8uf4q4af
412:       g.<q5116u-5cq8uc4&|:7c32&_7:|c32~49q5u-8dc>b-r
413:       u<q418c>fa<cq8e4q5u-8116fefr
414:       uq8c4&|:7c32&_6:|c32r8@E50,r8@u90@v8014c+
415: (t7)  K0o4q6c418c.u-6c16q8uf4q4af
416:       g.<q5116u-5cq8uc4&|:7c32&_7:|c32~49q5u-8dc>b-r
417:       u<q418c>fa<cq8e4q5u-8116fefr
418:       uq8c4&|:7c32&_6:|c32r8@E40,r8@u100@v90>14'a<c+'
419: (t8)  @u115K0o3r2q414cc8.q6@A100c16 @A100,,95,90,85,80,75,70
420:       q4fag8.<@A100q6u-8c16u@A100,,95,90,85,80,75,70c@A100
421:       ru-8116dc>b-r<q4u18c>fa<c @A100,,95,90,85,80,75,70u+8
422:       e4q6@A100u116fefru-8q7cr@A@E40,r*26>>@v90q814'a<a<c+'
423: (t9)  @u117@v90o2q614fcfc fcfc fcfc fcfa
424: (t10) wr1 r1 r1 r2.o6q6@u60a4
425: (t12) @Y26,40,100q3@Y29,40,65 @u50o114bbbb bbbb bbbb bbb@u65b
426: (t13) @u80o314aq2r2. r1 r1 r2.a
427: (t14) @v110q3o2116|:4@u50d8@u40dd@u50dZ35,30ddd@u50d8@u40dd
428:       @u50dZ35,30ddd:|
429: (t17) w116K0@u60@v95o4q6c4c8.cq8f4q4a8f8
430:       @u70g8.<@u110q5cq8c4&|:7c32&_5:|c32~35q6@u100dc>b-r
431:       <q4c8>f8a8<c8q8e4q6@u50fefr
432:       @u100q8c4&|:7c32&_5:|c32r4~35@u58q6c+4
433:
434: /B23       D－2
435: (t1)  w16q6dq4u+6c8>b-8<q6uc>u-6a ub-q4u+6a8g8q6uau-6f
436: (t2)  q6b-q4u+6a8g8q6uau-6f u@u75eq4u+6e8e8q6ufu-6c
437: (t6)  q6u+614dq4uc8>b-8<q6c>u-6a ub-q4a8g8q6uau-5f
438: (t7)  u+6q6'b-<d'uq4'a8<c''g8b-'q6'a<c'u-4'fa'
439:       u'eb-'u+4q4'e8a''e8g'uq6'fa'u-5'df'
440: (t8)  u+6q6'b-<b-<d'q4'a8<a<c''g8<gb-'uq6'a<a<c''f<fa'
441:       u'g<gb-'q4'e8<ea''e8<eg'q6'f<fa'u-4'd<df'
442: (t9)  b-a8g8af gf8e8fd
```

▶久しぶりにとことん使ってみたくなったものです。SX-BASIC。しかし時間が……。
古味　貴徳(20)千葉県

```
443: (t10) q6a4q4a8g8q6a4f4 e4q4e8e8q6f4f4
444: (t12) brbb brbb
445: (t13) |:u-10aru-10aua:|
446: (t14) r1 r1
447: (t16) wr1 @p30@u110@v50r1
448: (t17) q6d4q4c8>b-8<q6c4>a4 b-4q4a8g8q6a4f4
449:
450: /B24        D - 3
451: (t1)  @45q8@u90_13o5a2&|:7a16&_6:|a16 r1
452: (t2)  q8@u75o4a1 r1
453: (t6)  116@E60,@u100@v105q5|:3o4u-8a<ude|r:|o4u-8a8<uder
454:       |:3o5u-8euga|r:|o5u-8e8ugar
455: (t7)  116@E50,@u110@v95q6|:3o4u-8a<ude|r:|o4u-8a8<uder
456:       |:3o5u-8euga|r:|o5u-8e8ugar
457: (t8)  o2uq4'a<a<d''a<a<d'
458:       q5'a8.<a<d''a8<a<d'116q6u-3'a<a<d''a<a<d'r
459:       uq4'a4<a<c+''a4<a<c+'
460:       q5'a8.<a<c+''a8<a<c+'q6u-4'a<a<c+''a<a<c+'ru
461: (t9)  18aeaeae16ae16a aeaeae16ae16a
462: (t10) @u85132|:4efu-2efefu-2ef:| |:4ab-u-2ab-ab-u-2ab-:|
463: (t12) |:bbb8.b8.b8:|
464: (t14) @u50|:3ddd|r:|d8ddr |:3ddd|r:|d8ddr
465: (t16) o5e4&~30e4&@B0,8191,0~14e1& @B8191,4069,0e2
466: (t17) r1 116@u100q6|:3o4u-8euga|r:|o4e8gar
467:
468: /B25        D - 4
469: (t1)  w3w4@49r1 @Y1,99,64@Y1,32,76r2@E,25@H60@M20
470:       o3@u100@v80q6r418aa
471: (t2)  r1 r2.@H60@M20o4@u50q618aa
472: (t3)  wq6@u85o314d>a<d>a <d>a<d@u90@v80@Y1,99,64@Y1,32,76
473:       @E60,25o3@H60@M20@B-15018aa
474: (t4)  wq6@u65o414d>a<d>a <d>a<dr
475: (t6)  @u100@H90@M40@B0,-663,138q8a2.@B0,-663,8@M@u80@62_5
476:       @q1{d_4u-8c>_4u-8bagu-8fu-8ed}2^8r4.@57r4@B0
477: (t7)  @H90@M40_8@B0,-663,138q8'e2.a'@B0,-663,6@M@q1u-8_5
478:       {g_3f_5e_5u-8d_5u-8c}4 @B0r1
479: (t8)  o2q8'a2<a<f'&|:7'a16<a<f'&_8:|'a16<a<f' r1
480: (t9)  r1 r1
481: (t10) r1 r1
482: (t12) @u40bbbb bbbb
483: (t14) |:Z60,50,45,60,50,45d8ddd8ddZ45,40ddddZ50,55d8d8:|
484: (t16) @A80,75,70,65,60,50,45,40o6@B8191,-8191,0e*480
485: (t17) u-20@q1a2.u-15{g_5f_5e_5d_5c}4 r1@u110@v85
486:
487: /B26        D - 5
488: (t1)  <@q2d2^8q8dec t136@q2d4&_2|:3d16&_2:|d16r4~8>q6aa
489:       <@q2d2^8t135q8u+5efd u@q2a2.q8u+5b-a
490: (t2)  <@q2d2^8q8dec @q2d2r4>q6aa <@q2d2^8q8u+5efd
491:       u@q2a2.q8u+5b-a
492: (t3)  <@q2d2^8q8dec @q2d4&_2|:3d16&_2:|d16r4~8>q6u+4aa
493:       u<@q2d2^8q8u+5efd u@q2a2.q8u+5b-au
494: (t6)  r1 r1 r1 r1
495: (t7)  r1 r1 r1 r1
496: (t8)  r1 112o4@u125@v110r4u+4q5eucc>q8b8.&|:9b32&_8:|b32 r1
497:       <~70r4u+4q5eueeq8e-8.&|:9e-32&_8:|e-32
498: (t9)  14d>a<dd gdgg d>a<d>a <fcff8a8
499: (t10) r1 r1 r1 r1
500: (t12) bbbb bbbb bbbb bbbb
501: (t14) |:4Z60,50,45d8ddd8ddZ45,40ddddZ50,60d8d8:|
502: (t16) r2 @B0r1 r1 r1
503: (t17) r1 r1 r*190 <112o4r4q5eeeq8e-8.&|:9e-32&_8:|e-32
504:
505: /B27        D - 6
506: (t1)  w5w11u@q1g2^8u-5q8gfg ut130@Ma1 r1 w4t134r1 r1@p23@E,60
507: (t2)  u@q1g2^8q8gfg @M16a@u100@Y1,32,64@Y1,99,60o5
508:       q6'gb''g+<c''a<c+''b-<d''b<e-' <q8'c2e'>'a2<c'
509:       @u100|:15'a8<d'&_5:|'a8<d'
510: (t3)  @q1g2^8q8gfg @Ma1 @B0@Y1,99,60@Y1,32,64@E55,r1 r1 r*191
511: (t4)  w@u100@v105@p25o6q6116ee-dd-c>_3bb-aa-_gg-f_ee-_dd-
512:       c>b_b-aa-_gf+f e_e-d_c+r*47@p120
513: (t5)  wr1 @u90o2a*8_16Z80,85{aaaa ~2aaaaa aa~aaa aaa~aaa}*184
514:       Z85,90|:4{aaa~1aaa~2}*48:| r1 r1
515: (t6)  r1 @u95o416rq6'gb''g+<c''a<c+''b-<d''b<e-'
516:       <q8'c2e'>'a2<c' u+7@A110,100,95,90,85,80,70,60'a<d'*384
517: (t7)  r1 @u115o416rq6'gb<gb''g+<cg+<c'
518:       'a<c+a<c+''b-<db-<d''b<e-b<e-' <q8'c2e<ce'>'a2<ca<c'
519:       o3@58@E,80@u125@v68@p97|:12'a16<df+'&~2:|~10'a4<df+'
520:       @u90@A90,,85,80,75,70,65,60'a1<df' @p10 @62 @E,20
521: (t8)  r2~72r2 16o3@u85q6r'gb''g+<c''a<c+''b-<d''b<e-'
522:       <'c2e'>'a2<c' @E95,@u110@v120116o4q8|:q4d8.q6aq8g+8&
523:       |:5g+&_8:|g+~40|q5{ddd}4u-10_10:|r8@p10
524:       @Y1,32,61@Y1,99,64r16@E60,40r*10
525: (t9)  b-fb-f >{agg+aa+b}1 <c2>a2 <dada dada
526: (t10) r1 @u90o516rq6b<cc+de- r1 r1 r1
527: (t11) wr4@53@u80@v100@p50@B0@E100,30@Y1,99,66r2. r1 r1 r1 r1
528: (t12) w13bbbb {rbbbbb}1 b2b2 bbbb bbbb
529: (t13) wr1 16raaaaa a2a2 r1 r1
530: (t14) Z60,50,45d8ddd8ddZ45,40ddddZ50,60d8d8 @u60{drr}2r2 r1
531:       |:Z60,50,45d8ddd8ddZ45,40dddd|Z50,60d8d8:|Z40,45dddd
532: (t16) r1 @A@v105r1 r1 @p26o6@B8191,3000,0e1
533:       @p30@v80@A80,80,80,80,77,74,68,61@B8191,-4096,0e*288
534: (t17) r*194 q6o3@u110@v7516r'gb''g+<c''a<c+''b-<d''b<e-'
535:       <'c2e'>'a2<c' @E85,@v100o4|:q4d8.q6a16q8g+8&
536:       |:5g+16_8:|g+16~40|q5{ddd}4u-10_10:|r4@E50,
537:
538: /B28        E - 1
539: (t1)  t136@u100@v65o5116@q4|:4'df'|'eg''fa''eg':|'fa''eg''df'
540:       t-1|:4'ce'|'df''eg''df':|'eg''df''ce'
541: (t3)  @u110@v77o418q4d.d16u-10q3ddudfu+10q4a.r16
542:       ug.u-10a16q3gfeeu-10q4c.r16
543: (t4)  @u80     o318q4d.d16u-10q3ddudfu+10q4a.r16
544:       ug.u-10a16q3gfeeu-10q4c.r16
545: (t6)  @Ar1 r1
546: (t7)  18@v90@Y1,99,58@u100@A90,,,80,70,60,40o3q5d.d16
547:       u-10q3ddudf@A90,,,80,70,60,40u+10q8a.r16
548:       @A90q5ug.a16u-10q3gfeeu-10@A90,,,,80,70,50q8c.r16
549: (t8)  18@u120@v110@A110,,,100,90,80,60o3q5d.@A110d16
550:       u-10q3ddudf@A110,,,100,90,80,60u+10q8a.r16
551:       @A110uq5g.a16u-10q3gfeeu-10@A110,,,,100,90,70q8c.r16
552: (t9)  dada cgcg
553: (t11) o5@H36@M34q8d1 e1
554: (t12) @u40bbbb bbbb
555: (t13) r1 r1
556: (t14) |:Z60,50,45d8ddd8ddZ45,40ddddZ50,60d8d8:|
557: (t16) r2 r1@A
558:
559: /B29        E - 2
560: (t1)  |:4'df'|'eg''fa''eg':|'fa''eg''df'
561:       |:4'eg''fa''gb-''fa':|w2'g+b''a<c''a<c''a<c'
562:       'a<c'rq6'a<c''a<c' 'a<c+''a<c+''a<c+'
563:       r32@H48@M0t134@p28q5@u110@v98@Y1,$20,61r32a8a8
564: (t2)  wr2.@u85o4q5a8a8
565: (t3)  uq4d.d16u-10q3ddudfu+10q4a.r16
566:       @v80<q3cuq5c4q3dq5c4>u-10q3gub-
567:       q8a16&|:6a16&_5:|a16r4@Y1,99,58@E30,r*49
568: (t4)  uq4d.d16u-10q3ddudfu+10q4a.r16
569:       <q3cuq5c4q3dq5c4>u-10q3gub- q8a2r*97
570: (t6)  r1 @62r1116@u110@v65o4 u-20'g+b'u+10'a<c'u'a<c''a<c'
571:       q4'a8<c'q6'a<c''a<c' q6'a<c+''a<c+''a<c+'r@57r4
572: (t7)  @A90,,,80,70,60,40uq7d.@A90q5d16u-10q3ddudf
573:       @A90,,,80,70,60,40q8a.r16 @A80<q3c
574:       @A90,,,80,70,60,40q7c8.r16@A90q3d
575:       @A90,,,80,70,60,40q6c4>@A90u-10q3gub-
576:       @A90,,80,70,60,50,30q8a2r4@A@p115@Y1,99,64r4
577: (t8)  @A110,,,100,90,80,60uq7d.@A110q5d16u-10q3ddudf
578:       @A110,,,100,90,80,60q8a.r16 <@A100q3c
579:       @A110,,,100,90,80,60q7c8.r16@A110q3d
580:       @A110,,,100,90,80,60q6c4>@A110u-10q3gub-q8
581:       @A110,,100,90,80,60,40a2r4
582:       @A@p16r8@E30,@u110@v75r*26
583: (t9)  >b-<f>b-<f cgcg ffer
584: (t11) u+5f1 g1 a4&_8|:5a8&_8:|a8u@M
585: (t12) bbbb bbbb r2.@u78b
586: (t13) r1 r1 r2.q8@u95a4
587: (t14) |:Z60,50,45d8ddd8ddZ45,40ddddZ50,60d8d8:|
588:       @u78dddd d8dd dddrr4
589: (t16) @14@B0r1 @p65r1116@u105@v48o5 q6'g+b'u-6'a<c''a<c''a<c'
590:       q4u'a8<c'q6u-6'a<c''a<c' q6u'a<c+''a<c+''a<c+'rr4
591:
592: /B30        E - 3
593: (t1)  w10t+118<@M28u-5@q3d2^8udu+5eucq8
594:       u-5d4&|:7d16&_4:|d16~28>uu+5q6aa ut-1<@q2d2^8q6feu-5c
595:       q8ud4&|:7d16&_4:|d16~28t136r4
596: (t2)  18<u-5@q3d2^8udu+5euc u-5@q2d2.>uu+5q6aa
597:       u<@q2d2^8q6feu-5c u@q2d2.r4
598: (t3)  @59@u100@v48o2q6112'a2.<da'@q3u-8|:3'a<da':|
599:       uq6'a2.<da'@q3u-8|:3'a<da':|
600:       uq6'b-2.<fb-'@q3u-8|:3'b-<fb-':|
601:       uq6'b-2.<fb-'r8@49@Y1,99,64r8
602: (t6)  @E60,25@u100@v85o418r4q4a.u-10q6a16u<@H12@M45q8d2
603:       @Mr@q1f4d@M64q8a2 >@Mr4q4b-.u-10q6b-16<u@M45q8f2
604:       @Mr@q1f4d@M64q8b-2@M
605: (t7)  @E60,25@u115@v95o418r4q4a.u-10q6a16u<@H12@M45q8d2
606:       @Mr@q1f4d@M64u-8q8a2 >@Mr4q4b-.u-10q6b-16<u@M45q8f2
607:       @Mr@q1f4d@M64u-8q8b-2@M
608: (t8)  @A75,,,70,60,40,30o212q8'a.<da<d'u-8@q3|:3'a12<da<d':|
609:       uq8'a.<da<d'u-9@q3|:3'a12<da<d':|uq8'b-.<fb-<d'u-8@q3
610:       |:3'b-12<fb-<d':| uq8'b-.<fb-<d'r4
611: (t9)  dada dadc >b-<f>b-<f >b-<f>b-<f
612: (t10) wr1 r1 r1 @E60,r2.@73@u70@v83o4@q1b-8a8
613: (t12) |:4br2.:|
614: (t13) |:4a4r2.:|
615: (t14) |:4Z60,50,45d8ddd8ddZ45,40ddddZ50,60d8d8:|
616: (t16) r1 r1 @A@E60,30r1 @73@B0@u72@v100@p74r2.o6@q1b-8a8
617:
618: /B31        E - 4
619: (t1)  @M0r1 @v83@Y1,$20,64r2.o5q3b-a
620:       u-5q8t134g4.q4g@q3t130ug6b-6<u-5d6u
621: (t2)  r1 r2.o4q3b-a u-5q8g4.q4g@q3ug6b-6<u-5d6u
622: (t3)  @E60,@u65@v69o4q8'e-1gb-' 'd1fa' @59@Y1,99,58@E30,o3
623:       @u100@v50'e2g<d'q616'eg<d''eg<d''eg<d'o2116
624: (t6)  r1 18@E45,65@v80r2.o4q3b-a q8u-6g4.q4gu16@q3gb-<u-4d
625: (t7)  r1 18@E40,55@v100r2.o4q3b-a q8u-6g4.q4gu16@q3gb-<u-4d
626: (t8)  @E60,@u50o3@q1'e-1gb-' 'd1fa' @v90@A90,,,80,75,70,65,60
627:       @E30,o3@u116'e2g<d'16q6'eg<d''eg<d''eg<d'@A@v70o2116
628: (t9)  e-b-e-b- dada egb-g
629: (t10) @H32@M64u-5g4.g8u{gb-<u-5d}2> ua4&|:7a16&_2:|a16@Mr4 r1
630: (t12) @u40bbbb bbbb @u62br@u78{bbb}2
631: (t13) r1 r1 u-20a4r4u{aaa}2
632: (t14) |:3Z60,50,45d8ddd8dd|Z45,40ddddZ50,60d8d8:|@u60ddddd8dd
633: (t16) @H28@M64g4.g8g6b-6u-5<d6> ua4&|:7a16&_2:|a16@Mr4 @77r1
634:
635: /B32        E - 5
636: (t1)  @H24@M45t134q8116e2&|:7e&_4:|e@M
637: (t2)  q8e1
638: (t3)  q3'a4<a<d''a4<a<d'q6'a8.<a<d''a8<a<d'q6'a<a<d''a<a<d'r
639: (t6)  @u90116u-8>q5a<uder >u-6a<uder >u-8a<ude >a8<der
640: (t7)  @u100116u-8>q6a<uder >u-8a<uder >u-8a<ude >a8<der
641: (t8)  q3'a4<a<d''a4<a<d'q6'a8.<a<d''a8<a<d'q6'a<a<d''a<a<d'r
642: (t9)  a8e8a8e8a8e16a8e16a8
643: (t10) r2@E50,@p30r2
644: (t12) @u65b8b8b8b8b8b16b8b16b8
645: (t13) r1
646: (t14) @u60|:3ddd|r:|d8ddr
647: (t16) r1
648:
649: /B33        E - 6
650: (t1)  w17t135~22q7egaregaat136r4.@E55,@u105@v65116r8
651: (t2)  q7116egaregaar2
652: (t3)  q3'a4<a<c+'q6'a8.<a<c+'q8'a<a<c+'r4@49r4
653: (t6)  u-8eugu+8ar uegu+8aa ur2
654: (t7)  u-8eugu+8ar uegu+8aa ur4@A@p10@E,20@Y1,99,58r4
655: (t8)  q3'a4<a<c+'q6'a8.<a<c+'q8'a<a<c+'
656:       r4@A@p10@E60,40r*46
657: (t9)  a8e8a8e16a8.r8r4
658: (t10) o6@u100@v95116egaregaar2
659: (t12) b8b8b8b16b16r2
660: (t13) r2.>@u60116ee@u45e@u60e<@u95
661: (t14) dddrddddr4 @u80dd@u50_24d~24@u80d
662: (t16) @u100@v85@p70r4@E120,20@A85,80,77,74,71,68,65,62
663:       r*45@B8191,-4096,0o6e*291
664: (t17) w@u100@v75q7o4116u-8eugu+8aruegu+8aa r8@E60,25
665:       r16@Y1,99,64r16@Y1,$20,74r*46@A90,,80,70,60
666:
```

▶我が家の電気製品はシャープだけしかない。　久保　清光(41)北海道

```
667: /B34          E - 7
668: (t1)  w1lt138o5@q4|:4'df'|'eg''fa''eg':|'fa''eg''df'
669:       t136|:4'ce'|'df''eg''df':|'eg''df''ce'
670: (t6)  r1 r1
671: (t7)  @v90@u90@A90,,,,80,70,60,40o318q7d.@A90
672:       q5d16u-10q3ddudf@A90,,,,80,70,60,40u+10q7a.r16@A90
673:       q5ug.a16u-10q3gfeeu-10@A90,,,,,80,70,50q8c.r16
674: (t8)  @v110@u110@A110,,,,100,90,80,60o318q7d.@A110
675:       q5d16u-10q3ddudf@A110,,,,100,90,80,60u+10q7a.r16
676:       @A110uq5g.a16u-10q3gfee
677:       u-10@A110,,,,,100,90,70q8c.r16
678: (t9)  dada cgcg
679: (t11) w@v100@p74o5@M39d1 e1
680: (t12) @u40bbbb bbbb
681: (t14) |:Z60,50,45d8ddd8ddZ45,40ddddZ50,60d8d8:|
682: (t16) r1@B0
683: (t17) @u110@v90o4q4d8.q5d16u-10q3d8d8ud8f8u+10q4a8.r16
684:       uq4g8.q5a16u-10q3g8f8e8e8u-10q4c8.r16
685:
686: /B35          E - 8
687: (t1)  t134|:4'df'|'eg''fa''eg':|'fa''eg''df'
688:       |:'eg''fa''gb-''fa':|
689:       'df''eg''fa''gb-''g+b''a<c''b<d'<'ce'
690: (t6)  r1 r1
691: (t7)  @A100,,,,,90,80,60u+10q7d.r16ue.r16u+10q8f4uru-15q4d
692:       @A100,,,,,90,80,60uq8g.r16b-.r16@A90,,,80,70,60,50,30
693:       a4.r16@Y1,99,64@Ar32@E65,50r32
694: (t8)  @A120,,,,,110,100,80u+10q7d.r16e.r16u+10q8f4uru-15q4
695:       d*26 @A120,,,,,110,100,80uq8g.r16b-.r16
696:       @A110,,,100,90,80,70,50a4.r@A
697: (t9)  >b-<f>b-<f >ggaa
698: (t11) f1 g2a2
699: (t12) @u40bbbb bbbb
700: (t14) |:Z60,50,45d8ddd8ddZ45,40ddddZ50,60d8d8:|
701: (t16) r1 r1
702: (t17) q5d8.r16e8.r16u+10f4ur8u-15q4d*25
703:       uq5g8.r16b-8.r16a4r8@Y1,$20,64r*25@A
704:
705: /B36          F - 1
706: (t1)  w3w5w10t130@u110@v90r1 r4o3q318'b-<d'rr4<'ce'r
707: (t3)  w@v70r1 18r4o2q3'b-<d'rr4<'ce'r
708: (t5)  wa16r8.r2. r1
709: (t6)  @E70,60@u90@v80q3o5e8q8u-5e4q4uf8@q6u-5e4c4
710:       @H36@M32u@q20f2e2@M
711: (t7)  @p110@u115@v100q3o5e8q8u-5e4q4uf8@q6u-5e4c4
712:       @H36@M32u@q20f2e2@M
713: (t8)  @u120@v100r1 q6o2r4'b-4<b-<d'r4<'c4<ce'
714: (t9)  ~10a8r8r4r2 rb-r<c
715: (t10) w@C7r1 I8@62@u117@v100@p45@E50,35q8r1
716: (t11) r1 r1
717: (t12) w13r1 @u80rbrb
718: (t13) wr2@Y28,57,64@Y24,57,63r2 u-2214rarau
719: (t14) r1 @u112<a2a2>
720: (t16) @B0r1 r1
721: (t17) @u110@v65o4q3e8q8e4q4f8@q4e4c4 q6f2e2
722:
723: /B37          F - 2
724: (t1)  w2t135@45@u100@v70@p100o3'a1<d'& 'a2.<d'@p25@49o6q5
725:       @u110@v80{dc+c>u-5bb-au-5a-gg-u-8feu-8e-d}4
726: (t2)  wr1 r2.o5q5@u110{bb-au-5a-gg-u-5fee-u-8dc+u-8c>b}4
727: (t3)  @Y1,99,64r1 @58@u110@v100@p54@E30,50r1
728: (t5)  r1 r1
729: (t6)  @E55,@u105116q4d8rq6u+5aq7u@B0,-6668,83a-2@B0q6{ddd}4
730:       q4d8r16q6u+4a16uq8a-2.
731: (t7)  @E50,@u115116q4d8rq6u+5aq7u@B0,-6668,82a-2@B0q6{ddd}4
732:       q4d8r16q6u+4a16uq8a-2.
733: (t8)  @Y1,32,64q8@u110@v100o3'a<d'*12
734:       @u120@A30,40,50,65,60,50,40,30'a<d'*372
735: (t9)  _10d>a<d>a <d>a<d>a
736: (t10) 11o4@A50,55,60,65,70,75,80,85f+
737:       @A80,75,70,65,60,55,50,45f
738: (t11) ~8f+2.d8.u-8f+16 u @M50f2&|:7f16&_7:|f16
739: (t12) @u40bbbb bbbb
740: (t13) r1 @Y28,57,54r1
741: (t14) |:Z60,50,45d8ddd8ddZ45,40ddddZ50,60d8d8:|
742: (t16) o6@B8191,-4096,0@p20e*384
743: (t17) r1 @u110@v70r1o4
744:
745: /B38          F - 3
746: (t1)  t135@45@u90@v70@p100o3q8'a1<d'&
747:       |:15'a16<d'&_3:|'a32<d'r32~45@p25@49
748: (t3)  18o4|:q4d.q6a16q8g+4&|:7g+32&_5:|g+32|~35q5{ddd}4
749:       u-12:|r@59@u105@v60@p84r
750: (t5)  @44@u110r2@v90@p64@E50,70q8r2 r1
751: (t6)  r1 r1
752: (t7)  r1 r1
753: (t8)  'a<d'*384
754: (t9)  <d>a<d>a <d>a<d>a
755: (t10) o4@A50,55,60,65,70,75,80,f& @A80,75,70,65,60,55,50,45f
756: (t11) @Mr1 r1
757: (t12) bbbb bbbb
758: (t13) r1 @Y24,57,62r1
759: (t14) |:Z60,50,45d8ddd8ddZ45,40ddddZ50,60d8d8:|
760: (t16) @B8191,-4096,0e*384
761: (t17) |:q4d8.q6a16q8g+4&|:7g+32&_5:|g+32|~35q5{ddd}4u-15:|r4
762:
763: /B39          F - 4 (ラストブロックです！)
764: (t1)  w2w4116@u100@v80t133q8o5u+5ar8.q5u{ab-b}4<@M32q8c2@M
765:       >q5{ab-b}4<q8@M32c2@Mq6u+5{e-ce-}4
766:       d8r8>>t143u+5q6dq5d.t153r.t143q6dd8rr4
767: (t2)  w116@u80o4q8u+5ar8.q5u{ab-b}4<@M32q8c2@M
768:       >q5{ab-b}4<q8@M32c2@Mq6u+4{e-ce-}4
769:       d8r8>>u+5q6dq5d.r.q6dd8rr4
770: (t3)  12o3q8'a<d''b-<e-' 'a8<d'r8'b-<e-'u+5q6
771:       {'b-<e-''a<c''b-<e-'}4
772:       116'a8<d'r8>q6'a<d'q5'a.<d'r.q6'a<d'q5'a8<d'rr4
773: (t4)  wr1 r1 r1
774: (t5)  116r1 r1 o2r4q6'd<d'q7'd.<d'r.q6'd<d''d8<d'rr4
775: (t6)  116_20q8u-10ar8.u~20q5{ab-b}4<@M30q8c2@M
776:       >q5u-4{ab-b}4<q8@M30uc2@Mq4u-4{e-ce-}4
777:       ud8r8_12>>q5dd.r.q6dq5d8rr4
778: (t7)  116_20q8u-10ar8.u~20q5{ab-b}4<@M30q8c2@M
779:       >q5u-4{ab-b}4<q8@M30uc2@Mq4u-4{e-ce-}4
780:       ud8r@Y1,99,61r_12>>q5dd.r.q6dq5d8rr4
781: (t8)  @v80@A80,,,75,70,67,64,60@u95q8o2'a2<d<d''b-2<e-<e-'
782:       'a8<d<d'r8'b-2<e-<e-'@A@v80q6u+6
783:       {'b-<e-<e-''a<c<c''b-<e-<e-'}4116
784:       'a8<d<d'r8q6u+6'a<d<d''a.<d<d'r.'a<d<d'q5'a8<d<d'rr4
785: (t9)  <d2e-2 d4e-2{e-ce-}4 d8r8d16d16.r16.d16q5d8r16r4
786: (t10) r1 10r1 r1
787: (t11) r1 r1 r1
788: (t12) w15@u78b2b2 b8r8b2@u60{bbb}4 116b8r8bbr8bb8rr4
789: (t13) @u9512aa a8r8ar4 r1
790: (t14) r1 r2.@u60{ddd}4 d8r8ddr8dd8rr4
791: (t15) wr1 r1 r1
792: (t16) r1 r1 r1
793: (t17) @p30@v65@A65,,,60,55,52,49,45@u110q8o3'a2<d''b-2<e-'
794:       'a8<d'r8'b-2<e-'@A@v50q6{'b-<e-''a<c''b-<e-'}4116
795:       'a8<d'r8>q6'a<d'q5'a.<d'r.q6'a<d'q5'a8<d'rr4
796:
797: (p)
```

リスト2　宇宙戦艦ヤマト用カウンタ表示1（1～284行，/B15の前）

```
 1:00002358 00000000   2:000004F8 00000000   3:00001DB8 00000000   4:00000138 00000000
 5:00000A98 00000000   6:00001A28 00000000   7:00001728 00000000   8:000016F8 00000000
 9:000019C8 00000000  10:00000A08 00000000  11:000010C8 00000000  12:00002358 00000000
13:00000A68 00000000  14:000016F8 00000000  15:00000138 00000000  16:00000768 00000000
17:00001CC8 00000000
```

リスト3　宇宙戦艦ヤマト用カウンタ表示2（1～590行，/B30の前）

```
 1:00004A58 00000000   2:000011B8 00000000   3:00003BB8 00000000   4:000007F8 00000000
 5:00002358 00000000   6:00003528 00000000   7:00003228 00000000   8:000031F8 00000000
 9:000040C8 00000000  10:00001848 00000000  11:00001848 00000000  12:00004A58 00000000
13:00002B68 00000000  14:00003DF8 00000000  15:00001338 00000000  16:00001668 00000000
17:00002E08 00000000
```

リスト4　宇宙戦艦ヤマト用カウンタ表示3（1行～最後）

```
 1:00005AD8 00000000   2:00001C38 00000000   3:00004938 00000000   4:00000A38 00000000
 5:00002A18 00000000   6:000045A8 00000000   7:000042A8 00000000   8:00004278 00000000
 9:00005148 00000000  10:000025C8 00000000  11:00002208 00000000  12:00005AD8 00000000
13:000038E8 00000000  14:00004E78 00000000  15:00001578 00000000  16:000026E8 00000000
17:00003888 00000000
```

リスト5　プロジェクトA子


```
 1: /
 2: /プロジェクト・A子　オリジナル・サウンド・トラックより
 3: /
 4: /    S p a c e   s h i p   i n   t h e   d a r k
 5: /
 6: /      F o r   Z - M U S I C   V e r . 2 . 0 0
 7: /
 8: /              O h ! X  投稿バージョン
 9: /
10: /                 O P M + P C M 8
11: /
12: /           I p p i k o . W ( c ) 1 9 9 4
13:
14: (i)
```

▶3月号の「マッドストーカーX68」の記事を読んで急に興味がわいてきた（2月号までは興味がなかった）。買おうかな……。　木村　暢之(20)北海道

```
 15:
 16: / 音色設定
 17:
 18: (v70,0
 19: /          AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 20:            59, 15,  3,  0,  0,  0,  0,  0,  0,  3,  0
 21: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 22:            16,  0,  0,  3,  1, 18,  0,  7,  4,  0,  0
 23:            30, 14,  0,  3,  1, 33,  0,  3,  0,  0,  0
 24:            16, 14,  0,  3,  1, 20,  0,  1,  2,  0,  0
 25:            31,  0,  0,  6,  1,  0,  0,  2,  7,  0,  0)
 26: (v71,0
 27: /          AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 28:            53, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 29: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 30:            31,  0,  0,  2,  0, 23,  0,  4,  0,  0,  0
 31:            31, 17, 12,  8,  7,  0,  1,  2,  0,  0,  0
 32:            31, 17, 12,  8,  7,  0,  1,  3,  0,  0,  0
 33:            31, 17, 12,  8,  7,  0,  1,  4,  0,  0,  0)
 34: (v73,0
 35: /          AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 36:            61, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 37: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 38:            31,  5,  7,  2,  0, 22,  0,  2,  0,  0,  0
 39:            25, 12,  1,  8,  3,  0,  0,  1,  0,  0,  0
 40:            25, 12,  1,  8,  3,  0,  0,  4,  0,  0,  0
 41:            25, 12,  1,  8,  3,  0,  0,  2,  0,  0,  0)
 42: (v74,0
 43: /          AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 44:            62, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 45: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 46:            20,  0,  0,  3,  0, 44,  1,  4,  2,  0,  0
 47:            12,  5,  2,  7,  0,  0,  1,  2,  2,  0,  0
 48:            12,  5,  2,  7,  0,  0,  1,  4,  7,  0,  0
 49:            12,  5,  2,  7,  0,  0,  1,  1,  7,  0,  0)
 50: (v75,0
 51: /          AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 52:            62, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 53: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 54:            24,  4,  0,  0,  0, 15,  0,  0,  7,  0,  0
 55:            20, 20,  3,  5,  5,  6,  0,  2,  7,  0,  0
 56:            12,  5,  3,  5,  2,  4,  0,  1,  3,  0,  0
 57:            12, 10,  3,  5,  2,  4,  0,  2,  7,  0,  0)
 58: (v76,0
 59: /          AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 60:            44, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 61: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 62:            14,  0,  0,  0,  0, 19,  0,  1,  7,  0,  0
 63:            13,  3,  0,  5,  2,  0,  0,  1,  7,  0,  0
 64:            14,  0,  0,  0,  0, 25,  0,  1,  3,  0,  0
 65:            13,  3,  0,  5,  2,  0,  0,  1,  3,  0,  0)
 66: (v77,0
 67: /          AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 68:            38, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 69: /          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 70:            31,  0,  6,  3,  0, 15,  0,  2,  0,  0,  0
 71:            31, 10,  9,  7,  5,  0,  0, 12,  0,  0,  0
 72:            31, 10,  9,  7,  4,  4,  0,  1,  0,  0,  0
 73:            31,  8,  5,  7,  3,  4,  0,  0,  0,  0,  0)
 74:
 75: / PCMデータ 読み込み
 76: .ADPCM_BLOCK_DATA=SPACE.ZPD
 77:
 78: / M u s i c   T r a c k   A s s i g n
 79: (m1,5000)
 80: (m2,5000)
 81: (m3,5000)
 82: (m4,5000)
 83: (m5,5000)
 84: (m6,5000)
 85: (m7,5000)
 86: (m8,5000)
 87: (m9,5000)
 88: (m10,5000)
 89: (m11,5000)
 90: (m12,5000)
 91: (m13,5000)
 92: (m14,5000)
 93: (m15,5000)
 94: (m16,5000)
 95: (a1,1)
 96: (a2,2)
 97: (a3,3)
 98: (a4,4)
 99: (a5,5)
100: (a6,6)
101: (a7,7)
102: (a8,8)
103: (a9,9)
104: (a9,10)
105: (a9,11)
106: (a9,12)
107: (a9,13)
108: (a9,14)
109: (a9,15)
110: (a9,16)
111:
112: / T e m p o  設定
113: (o164)
114:
115: / 演奏  T r a c k   D a t a
116:
117: / M e l o d y  1
118:
119: (t1) @12 r @k3 @73 p1 o4 v14 112
120: (t1) r1 r1  r1 r1 r1 r1  r1 r1 r1 r1 |:31 r1 :|  r2{ggg}
2
121: (t1) |:4 a8.&(a>a)12<r2. r2 | {ggg}2 :| {aaa}2 |:4 b8.&(
b>b)12<r2. r2 | {aaa}2 :| < {ccc}2
122: (t1) |:4 d8.&(d>d)12<r2. r2 | {ccc}2 :| {ddd}2 |:4 e8.&(
e>e)12<r2. r2 | {ddd}2 :| r2
123: (t1) @77 p3 o6 v14 |:2 e1& e2d4e4 f1& f2e2 d1& d2>g4<g4
f4&f6ee2& e2d2 c1& c2d4e4 f1& f2g2 d2.c4  {>b<cd}1 > a1& a1 :|
124: (t1) @76 v6 o4 |:16 r1 :| a1& a1 f1& f1 g1& g1 <c1& c2>b
2 < a1& a1 f1& f1 g1& g1 a1& a1 @77 p3 v14
125: (t1) o5 |:2 d1& d2e4f4 c1&c2c2 | e1& e2f4g4 d1& d2c2 :|
d1 c1 >b1& b1
126: (t1) o6 |:2 e1& e2d4e4 f1& f2e2 d1& d2>g4<g4 | f4&f8e8e2
& e2d2 c1& c2d4e4 f1& f2g2 d2.c4  {>b<cd}1 > a1& a1 :| < f4f6e12
e2& e4.&e24f24e24 d2 c1& c2d4e4 f1& f2g2 d2.c4  >b4.<c4.d4 > a1&
 a1
127: (t1) o6 e1& e2d4e4 f1& f2e2 d1& d2>g4<g4 f4&f8e8e2& e2d2
 c1& c2d4e4 f1& f2g2 d2.c4  {>b<cd}1 > a1& a1& a2
128:
129: / M e l o d y  2
130: (t2) @12 r @k-3 @73 p2 o4 v14 112 r32
131: (t2) r1 r1  r1 r1 r1 r1  r1 r1 r1 r1 |:31 r1 :|  r2{ggg}
2
132: (t2) |:4 a8.&(a>a)12<r2. r2 | {ggg}2 :| {aaa}2 |:4 b8.&(
b>b)12<r2. r2 | {aaa}2 :| < {ccc}2
133: (t2) |:4 d8.&(d>d)12<r2. r2 | {ccc}2 :| {ddd}2 |:4 e8.&(
e>e)12<r2. r2 | {ddd}2 :| r4...
134: (t2) @77 p2 r4 o6 v13 |:2 e1& e2d4e4 f1& f2e2 d1& d2>g4<
g4 f4&f6ee2& e2d2 c1& c2d4e4 f1& f2g2 d2.c4  {>b<cd}1 > a1& a1 :
|
135: (t2) @76 p2 v5 o4 |:16 r1 :| a1& a1 f1& f1 g1& g1 <c1& c
2>b2 < a1& a1 f1& f1 g1& g1 a1& a1 @77 p2 v13
136: (t2) o5 |:2 d1& d2e4f4 c1&c2c2 | e1& e2f4g4 d1& d2c2 :|
d1 c1 >b1& b1
137: (t2) o6 |:2 e1& e2d4e4 f1& f2e2 d1& d2>g4<g4 | f4&f8e8e2
& e2d2 c1& c2d4e4 f1& f2g2 d2.c4  {>b<cd}1 > a1& a1 :| < f4f6e12
e2& e4.&e24f24e24 d2 c1& c2d4e4 f1& f2g2 d2.c4  >b4.<c4.d4 > a1&
 a1
138: (t2) o6 e1& e2d4e4 f1& f2e2 d1& d2>g4<g4 f4&f8e8e2& e2d2
 c1& c2d4e4 f1& f2g2 d2.c4  {>b<cd}1 > a1& a1& a2
139:
140: / C o d e  1
141: (t3) @k3  @76 @q6 p1 v08
142: (t3) r1 r1  r1 r1 r1 r1  r1 r1 r1 r1
143: (t3) o5 |:2 |:2 e1& e1 f1& f1  d1& d1 | e1& e1 :| c1& c1
 :|
144: (t3) v9 |:6 r1 :| a1& a1  |:6 r1 :| b1& b1  < |:6 r1 :|
d1& d1  |:6 r1 :| e1& e1
145: (t3) o5 |:2 |:2 e1& e1 f1& f1  d1& d1 | e1& e1 :| c1& c1
 :|
146: (t3) o3 p3 |:2 a1& a1 f1& f1 | g1& g1 < c1& c2 > b2 :|
< d1& d1 e1& e1 >
147: (t3) v10 |:2 a1& a1 f1& f1 g1& g1 | <c1& c2>b2 :| a1& a1
148: (t3) o3 d1& d1 c1& c1  e1& e1 d1& d2f2  b-1& b-1 f1& f1
 g1& g1& g1& g1
149: (t3) o4 |:2 a1& a1 _6 f1& f1  g1& g1 ~6 | <c1& c2>b2  a1
& a1 f1& f1 g1& g1 a1& a1 :| _3 c1& c2>b2  |:2 a1& a1 f1& f1 g1&
 g1 | a1& a1 ~6 :| <c1& c2>b2  a1& a1 f1& f1 g1& g1 a1& a1& a2
150:
151: / C o d e  2
152: (t4) @k-3 @76 @q6 p2 v08
153: (t4) r1 r1  r1 r1 r1 r1  r1 r1 r1 r1
154: (t4) o5 |:2 |:2 e1& e1 f1& f1  d1& d1 | e1& e1 :| c1& c1
 :|
155: (t4) v9 |:4 r1 :| |:3 e1& :| e1  |:4 r1 :| |:3 g-1& :| g
-1  |:4 r1 :| |:3 a1& :| a1  |: 4 r1 :| |:3 b1& :| b1
156: (t4) o5 |:2 |:2 e1& e1 f1& f1  d1& d1 | e1& e1 :| c1& c1
 :|
157: (t4) o4 |:16 r1 :| p3 |:2 a1& a1 a1& a1 | g1& g1 g1& g2g
2 :| b1& b1 < c1& c1
158: (t4) o5 d1& d1 c1& c1  e1& e1 d1&d2f2&  f1& f1 f1& f1  d
1&d1 g1& g1
159: (t4) o5 |:3 e1& e1 f1& f1 d1& d1 e1& e2d2   e1& e1 f1& f
1 d1& d1 | e1& e1 :| > a1& a1& a2
160:
161: / C o d e  3
162: (t5) @k3  @76 @q6 p1 v09
163: (t5) r1 r1  r1 r1 r1 r1  r1 r1 r1 r1
164: (t5) o5 |:2 |:2 c1& c1 c1& c1 > b1& b1 < |  c1& c1  :| >
 a1& a1 < :| v10
165: (t5) v11 o4 r1 r1 |:5 a1& :| a1  r1 r1 |:5 b1& :| b1  <
r1 r1 |:5 d1& :| d1  r1 r1 |:5 e1& :| e1
166: (t5) o5 |:2 |:2 c1& c1 c1& c1 > b1& b1 < |  c1& c1  :| >
 a1& a1 < :| v10
167: (t5) o4 |:16 r1 :| p2 e1& e2g2 f1& f1 d1& d1 e1& e2d2  e
1& e1 f1& f1 g1& g1 a1& a1
168: (t5) o4 p2 a1& a1 g1& g1 g1& g1 g1& g1 < d1& d1 c1& c1 >
b1& b1 < d1& d1
169: (t5) o5 |:3 |:2 c1& c1 c1& c1 > b1& b1 | < c1& c2>b2< :|
 | c1& c1 :| a1& a1& a2
170:
171: / C o d e  4  &  V o i c e  1
172: (t6) @k-3 @76 @q6 p2 v09
173: (t6) r1 r1  r1 r1 r1 r1  r1 r1 r1 r1
174: (t6) o4 |:2 |:2 a1& a1 a1& a1 g1& g1 | g1& g1 :| e1& e1
:|
175: (t6) v11 |:7 e1& :| e1  |:7 g-1& :| g-1  |:7 a1& :| a1
|:7 b1& :| b1 v10
176: (t6) o4 |:2 |:2 a1& a1 a1& a1 g1& g1 | g1& g1 :| e1& e1
:|
177: (t6) @75 @s10 @h16 @m16 @q4 o5 v13 |:2 c1& c1 c1& c1 > b
1& b1 | < c1& c2>b2<
178: (t6) |:2 c1& c1 c1& c1 d1& d1 e1& | e1 :| e2d2 :| c1& c1
179: (t6) @q0 o5 r2d4>b-4 {fd>b-}2< |:2 {fd>b-f<}4 :|  r2c4>g
4 e1
180: (t6) <<  r2g4d4 > b-1 g1& g2<f2& f4r4f4d4 >b-4f4<f2& f4r
4@q4c8>a8g8@q0f8& f1
181: (t6)r2<{dr}8>b8&b4 a8g8&g4a4b4< g1& g1
182: (t6)o5 |:3 c1& c1 c1& c1 > b1& b1 < e1& e2d2  e1& e1 f1&
 f1 d1& d1 | c1& c1 :| > a1& a1& a2
183:
184: / エレキ・ギター  &  V o i c e  2
185: (t7) @12 r @k0  @76 @q6 p3 v8 112
186: (t7) r1 r1  r1 r1 r1 r1  r1 r1 r1 r1
187: (t7) |:32 r1 :| |:20 r1 :|
188: (t7) @70 o4 v13 @s5 @h0 @m p2 r4drdf4d6c>  a&b-&gf6e&e4&
@m48e8&(e>e)24  << @m r4p3ffrf4rr>f< f6>f<f6f&f4r4  |:5 b16&<d16
d16> :| b16& <d&>b&<d&@m48d2&d8&(d>a)24
189: (t7) o5 @m v14 112 r4dd-c>a6&(a,c)48c(c,g)12&g8. p2 e2&@
```

▶ジョルトコーラはとても美味しい。カラムーチョとよくあうんだな，これが。
宗京 邦和(21)奈良県

```
m48e4&(e>e)24r8 < @m p3 |:2 (e,b)16&b<d | d6e& e8.&(e>b)12 :| <
e6e& e4.&(e>e)24
   190: (t7) o5 @m r6(a<e)16&e2e@k-10e@k-21e @k-31e@k-42e@k-54e
@k0e-@k-10e-@k-20e-_1 @k-31e-_1@k-42e-_1@k-54e-_1 @k0d_2@k-10d_2
@k-21d  @k-31d_2@k-42d_2@k-54dr2.
   191: (t7) |:13 r1 :| @k0 @70 v14 o4 l12 r4r(c,e)16&ee6ee6e e&
dc>a6g@h12@m48@s5a4.&(a>a)24 |:4 r1 r1 r1 r1 :|
   192: (t7) @75 p1 @s10 @h16 @m16 @q4 o4 v13 |:2 a1& a1 a1& a1
g1& g1 | g1& g2g2
   193: (t7) |:2 a1& a1 a1& a1 b1& b1 <c1& | c1> :| <c2>g2 :| a1
& a1
   194: (t7) o5 r2d2& d1 r2c2& c1  r2g2& g1 g1& g1 f1& f1 c1& c1
 d1& d1 g1 d1
   195: (t7)o5 |:3 c1& c1 > a1& a1 g1& g1 <c1& c2>b2<  c1& c1 c1
& c1 > b1& b1 | a1& a1 :| > a1& a1& a2
   196:
   197: / B a s s
   198: (t8) @12 r @71 q1 l12 v14
   199: (t8) r1 r1
   200: (t8) o1 |:8 |:12 a :| :|
   201: (t8) o1 |:2 |:2 |:2 |:12 a :| :| |:2 |:12 f :| :| |:2 |:
12 g :| :| |  < |:12 c :| |:6 c :| > |:6 g :| :| |:2 |:12 a :| :
| :|
   202: (t8) |:4 |:18 a :| | |:6 g :| :| |:6 a :|  |:4 |:18 b :|
 | |:6 a :| :| < |:6 c :|
   203: (t8) |:4 |:18 d :| | |:6 c :| :| |:6 d :|  |:4 |:18 e :|
 | |:6 d :| :| |:6 e :|
   204: (t8) o1 |:2 |:2 |:2 |:12 a :| :| |:2 |:12 f :| :| |:2 |:
12 g :| :| |  < |:12 c :| |:6 c :| > |:6 g :| :| |:2 |:12 a :| :
| :|
   205: (t8) o1 q1 l8 |:2 |:2 |:16 a :| |:16 f :| |:16 g :| | <
|:12 c :| > |:4 b :| :| |:16 a :| :|
   206: (t8) o2 |:16 d :| |:16 c :| |:16 e :| |:12 d :| |:4 f :|
  |:16 b- :| |:16 f :| |:32 g :|
   207: (t8) o1 l8 |:3 |:2 |:16 a :| |:16 f :| |:16 g :| | < |:1
2 c :| > |:4 b :| :| |:16 a :| :|
   208:
   209: / B a s s   D r u m
   210: (t9) @2 l12 @v50 o3
   211: (t9) |:5 c6c |:6 c4 :| c6c :|
   212: (t9) |:16 c6c |:5 c4 :| | c4c6c :| {ccc}2
   213: (t9) |:16 c6c |:5 c4 :| {ccc}2 :|
   214: (t9) |:16 c6c |:6 c4 :| c6c :|
   215: (t9) |:48 r4c2r4 r1 :|
   216:
   217: / S n a r e   D r u m
   218: (t10) @2 l12 @v60 o3
   219: (t10) |:5 |:4 r4d6r :| :|
   220: (t10) |:3 |:16 |:4 r4d6r :| :| :|
   221:
   222: / H i - H a t
   223: (t11) @2 l12 @v70 k1 o3
   224: (t11) |:10 _32f¯32f_32f¯32 f_32f¯32f fff f_32f¯32g :|
   225: (t11) |:3 |:32 _32f¯32f_32f¯32 f_32f¯32f fff f_32f¯32g :
| :|
   226: (t11) @3 k0 @v56 o3 l8 |:48 |:16 e :| :|
   227:
   228: / P e r c .
   229: (t12) @2 l12 @v60 @f3
   230: (t12) o3 c2r2 r2.o6 @v80 @f3 d4 @v60 o3 @f3 |:2 c2 |:3 r
2 :| :|  |:2 @2 o3 @f3 @v60 c2 @3 o1 @f4 b4r4 @2 o4 @v80 r|:5 c#
_10 :| @v60 @3 o1 b4r4  :|
   231: (t12) |:3 |:16 @2 o3 @f3 @v60 c2 @3 o1 @f4 b4r4 @2 o4 @v
80 r|:5 c#_10 :| @v60 @3 o1 b4r4  :| :|
   232: (t12) |:48 @2 o3 @f3 c2 @3 o1 @f4 b2 r1 :|
   233:
   234: / オーケストラ・D r u m
   235: (t13) @2 o5 l12 @v32 @f3
   236: (t13) r1 r1  r1 r1 r1 r1  r1 r1 r1 r1
   237: (t13) |:16 |:3 d4ddd :| dddddd :|
   238: (t13) |:16 |:2 r1 :| :|
   239: (t13) |:15 |:3 d4ddd :| dddddd :|
   240: (t13) |:2 d4¯1 d¯2d¯1d¯1 :| d4¯3 |:9 d¯3 :|
   241:
   242: / オーケストラ・ヒット
   243: (t14) @1 l12 @v70
   244: (t14) r1 r1  r1 r1 r1 r1  r1 r1 r1 r1
   245: (t14) o0 @f2 |:2 e1& e2d4e4 f1& f2e2  d1& d2g4g4 f4&f6ee
2& e2d2
   246: (t14) c1& c2d4e4 f1& f2g2 d2.c4 @f0 b4.@f2c4.d4 @f0a1& |
 a1 @f2 :| @f2 a2{ggg}2
   247: (t14) |:4 a1& a2 | {ggg}2 :| {aaa}2  |:4 |:3 b2_20 :| ¯6
0 | {aaa}2 :| @f4 {ccc}2
   248: (t14) |:2 |:3 d4&d6 | _20c :| r ¯40 {ccc}2 :| |:2 |:3 d2
_60 :| ¯60 | {ccc}2 :| {ddd}2 · |:4 |:3 e2_20  :| | ¯60 {ddd}2 :|
 e2
   249:
   250: (p)
```

リスト6　プロジェクトA子用音色コンフィグファイル

```
.ADPCM_BANK1

.O0C   OH2C.PCM,p12
.O0C#  OH2C#.PCM,p12
.O0D   OH2D.PCM,p12
.O0D#  OH2D#.PCM,p12
.O0E   OH2E.PCM,p12
.O0F   OH2F.PCM,p12
.O0F#  OH2F#.PCM,p12
.O0G   OH2G.PCM,p12
.O0G#  OH2G#.PCM,p12
.O0A   OH2A.PCM,p12
.O0A#  OH2A#.PCM,p12
.O0B   OH2B.PCM,p12
.O1C   OH2CC.PCM,p12

.ADPCM_BANK2

.O3C   TECK.PCM
.O3D   OUCHS_.PCM
.O3F#  CH1.PCM
.O3G#  OH1.PCM

.O4C#  RIMSHOT.PCM,p2

.O5D   TR808SD.PCM

.O6D   BOS_SN.PCM

.ADPCM_BANK3

.O1B   JBELL1.PCM

.O3E   SHAKER.PCM
```

リスト7　プロジェクトA子用カウンタ表示

```
 1:000097E2 00000000   2:00009812 00000000   3:000097E0 00000000   4:000097E0 00000000
 5:000097E0 00000000   6:000097E0 00000000   7:000097E2 00000000   8:00009782 00000000
 9:00009780 00000000  10:00004F80 00000000  11:00009780 00000000  12:00009780 00000000
13:00004F80 00000000  14:00003780 00000000
```

(進)の「ちょっといいですかぁ?」

◆宇宙戦艦ヤマト

これはSC-55を存分に生かした力作です。リストは非常に長いものとなっていますが，聴ける環境にある人はぜひ入力してみてください。

音源の制御(フィルタの開閉やエンベロープの調節)から，ベロシティやテンポの流れ，音の切れなど，よく練られた作品です。苦労が音に出ているのがわかるでしょうか。

逆にいうと，いかにMIDI音源といえども，普通に鳴らしただけでは味が出にくいということでしょうね。

ここまでくると，SC-55の薄いベースや，張りのない金管にもの足りなさを感じてしまうかもしれません(私は感じた)。

ひと言つけ加えるとすると，テンポの揺れがちょっと激しいかなということと，全体的な音量(強弱)の変化がさらに欲しいということです(特に最後はもっと盛り上がってもいいかも)。

パソコン通信でも活躍しておられる早坂さんだけに，データは参考になるはずです。NRPN(@Yコマンド)による音色パラメータの変更，ベロシティシーケンスの効果的な使い方など，いろいろ研究してみてください。

最後にちょっと注意点。これはSC-55専用のようですから，SC-33やSC-55mkⅡなど，SC-55と完全互換でない機種で演奏した場合は，ニュアンスが変わって聴こえることもありえます(ただし，そんなひどい演奏にはならないと思いますが)。ご了承ください。

◆プロジェクトA子

なかなかうまくまとめられていて，手慣れた印象を受けます。いかにもコンピュータ音楽的な音色もハマっていますし，バランスもなかなかのもの。

柔らかな音色中心で音程も高いため，ところどころ薄く感じられることもあります。音が少なくなるところでは，空いているチャンネルを適度に割り当てるといいかもしれません(やり過ぎは禁物ですが)。

それから前半，オーケストラヒットをメロ的に使っているパートがあるのですが，これは抑え目にしたほうがいいですね。

近頃は内蔵音源の投稿が少なくなって私は寂しいです。ほかの皆さんも頑張ってください。

それではまた。　　　　　　　　(進藤慶到)

(善)のゲームミュージックでバビンチョ

西川善司

PC-9821AP2を買い，Windowsを導入した。インストールに苦労した。ウィンドウアクセラレータMGA-IIと486D X2（66MHz）のパワーを駆使しても，Windowsのパフォーマンスは X68030上のSX-WINDOWのそれ以下であった。WordPerfect for Windowsで英文を書いていたら，よく暴走して文章が消えた。私は苦笑いが絶えなかった。WindowsのCMのキャッチコピー「笑ってお仕事」はこういう意味だったのか……。ちなみにこの原稿はシャーペン.Xで書いている。

GAMADELIC!!

去る2月5日，原宿ルイードで「GAMADELIC・DELICIOUS SELECTION」の発売を記念して，招待されたファンだけが参加できるというプレミアムライブが行われた。ものすごい盛況ぶりで会場は超満員。

スクリーンに怪しげな映像が映し出され，「GAMADELICのテーマ」に乗って颯爽とメンバーが登場。そのベストアルバムに収録されている曲をすべて演奏(!)。さらに，ゲームのほうは未公開という対戦格闘ゲーム「FIGHTER'S HISTORY DYNAMITE」，対戦円盤投げゲーム(?)「FLYING POWER DISC」の曲をファンのために初公開。アンコール2曲も含めなんと全部で16曲を快演。途中のメンバーの宝物をプレゼントする抽選コーナーも好評。ファンは大満足で帰路についた。

あ，それと私ももらった来場者全員プレゼントの「FLYING POWER DISC」フリスビーはうちの犬の宝物になっている。

*　　　*

●FIGHTER'S HISTORY DYNAMITE・FLYING POWER DISC/DATA EAST GAMADELIC
CD:PCCB-00149　1,500円(税込)
ポニーキャニオン　3/18発売

というわけで，1枚目はこれ。カプコンとのゴタゴタをよそに，いきなり問題のゲームの続編を発表。しかも今度はNEO・GEOから。つまり「はいななむにゃにゃー」や「肛門科！」の叫びが家庭で楽しめることになるわけ。いまからわくわくしちゃうよ。

んで，このCDはもうひとつNEO・GEOゲームの「FLYING POWER DISC」も収録。両ゲームともオリジナルサウンドは効果音も含めて完全収録。1,500円のCDで収録時間74分とはポニーキャニオンたら，ずいぶんと太っ腹。

このCD，アレンジバージョンが2曲収録されている。トラック1の「DYNAMITE」はボコーダーを効果的に使っていて面白い。後半のMr.Kのラップも絶妙。ライブの余韻も残っているせいで思わず熱くなってしまった。

お勧め度　8

●CGMV Virtua Fighter
VHS:TYVY-5001　3,800円(税込)
東芝EMIユーメックス　発売中

ハウスやダンスミュージックのプロモーションビデオクリップを意識した構成で，攻略系のビデオとは一味違った趣が感じられる。で，アメリカンやユーロな感じで統一してあるのかと思ったらいきなり被ってくるインチキDJ風のナレーション。日本語を巻き舌で話すのはヤ・メ・ロ……。

とはいえこのビデオ，ゲームに入れ込んだ人でなくても楽しめるだけの映像的，音響的な魅力は十分備わっている。攻略ビデオというよりバーチャファイターを音と絵で綴ったものといえる。

しかし，みんな勝ちポーズの顔が怖い。

お勧め度　8

●Virtua Fighter　B-univ
CD:TYCY-5386　3,000円(税込)
東芝EMIユーメックス　3/23発売

なんか，解散してから，昔ながらのセガの味が帰ってきた気がする。みなさんのご期待通りいい味出してます。今回の2ndアルバムも前半がアレンジ，後半がオリジナルサウンドという構成。期待の効果音集は残念ながらなし。「10年早いんだよぉぉ」をエラービープにしたかった私の野望は……。

アレンジは文句なしの出来。全曲ダンサブルなハウス系。聴いてて元気が出てくるようなカッコよさ。でも，トラック3のバーチャファイターのテーマ「君はきわめろ」は日本語ラップで賛否両論かな。私は日本語ラップ撲滅委員会のメンバーだから「ダメ」。聴いてて「祖父マップ」の店内放送を思い出した。ちなみにこの曲はカラオケバージョンが収録されているので，肩を揺らしながら巻き舌で「世界のファイター集まRRRればRRR」と歌おうが歌うまいが君の勝手だ。

お勧め度　9

●MIDI POWER Ver.4.0
コナミ矩形波倶楽部
CD:KICA-7634　2,800円(税込)
キングレコード　3/24発売

最初はX68000版に移植されたコナミゲームのサウンドトラック的な趣向だったこのシリーズ。「Ver.3.0」では「グラディウス3」が，X68000には移植されていないタイトルなのに収録されており，しかも「X68000で演奏している」というようなことがクレジットしてあって，ファンは「グラ3，X68000に移植されるのかなぁ」と淡い期待を寄せたものだ(未だに淡いままだが)。で，今回は「XEXEX」，「サンダークロス」をいつも通りSC-55で再現させている。うーん，この2タイトルもX68000に移植してほしいソフトの上位人気作だけに期待は高まるが。さて，収録曲いつもながらあっぱれな完成度です。

お勧め度　8

FIGHTER'S HISTORY DYNAMITE

Virtua Fighter

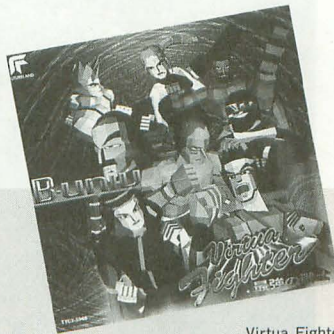
(で)のショートプロぱーてぃ――その55

時代はミックスなのだ!

illustration : T.Takahashi

Komura Satoshi 古村 聡

今月のショートプロはX-BASICにC言語,アセンブラと盛りだくさん。どれも短いのでちょちょいと入力して試してみましょう。(で)さんはいろいろと考えすぎて頭のなかがミックスになったのか,プログラム風まかせは残念ながらお休みです。

ビールのつまみにミックスナッツ。うまいんだな,これが。ひとつの袋にいろいろな種類のナッツが入っているから飽きるってことがなくていいんですよね。考えてみると,ミックスと名前につくものは,たとえばミックスベジタブル,ミックスジュース,ゆみみみっくす,ミックスナッツ,日経MIX,どれもいろいろ入っていて,飽きのこないいいものが多いですよね。えっ,関係ないものがまじっている? 気にしない気にしない。

ところで最近,パソコンに限らず,ゲームのジャンルによって遊び方がかなり固定されてきていると思いませんか? シューティングゲーム,格闘ゲーム,ロールプレイングゲーム,パズルにデジタルコミック。どのジャンルも枠をはみ出すものがなかなかないんですよね。それにいくらいいゲームが多くても,同じようなゲームばかりだと飽きるでしょ。だったらミックスしてしまえばいいのだ!

たとえばロールプレイング風の格闘ゲーム。敵を倒して経験値を稼ぐと昇龍拳や波動拳が使えるようになるんです。いいと思いません? 問題は技を使えるようになっても出るとは限らないけど……。人間が技の出し方を覚えてないとね。

それじゃ,パズルな格闘ゲーム。いい位置で百裂ハリ手をかまさないと面クリアできないとか。だめ? ついでに格闘ゲームのデジタルコミック。敵を倒すとデジコミが見られる! それでもダメなら,逆にしてデジタルコミックな格闘ゲーム! マウスのボタンを押すとケンやリュウが昇龍拳するところが見られる!

えっ,1年間休暇やるから頭冷やしてこいって,そんな……。

小粋なパズルだ

今月の1本目はジャンルミックスなゲームってことで,アクション要素の混じったパズルゲームです。ちょいと小粋なリアルタイムパズル(?)ゲーム,SOLIX.BASです。どうぞ。

SOLIX.BAS for X680x0

(X-BASIC コンパイル推奨)

大阪府 西村志郎

このプログラムはX-BASICで書かれています。BASICコンパイラでコンパイルすることを前提としていますので,Cコンパイラ PRO-68Kを使って,

```
A>CC SOLIX.BAS
```

としてプログラムをコンパイルしてください。その際,warningのメッセージがいくつか出ますが,気にしなくても大丈夫。エラーメッセージが出てきた場合は打ち間違いですから,エラーメッセージに従って間違えている部分を直してコンパイルしてくださいね。

では遊び方。このゲームの操作にはマウスを使用します。ルールは,赤,青,紫の玉がリアルタイムで出てきますので,この玉を消してください。縦4×横4マスのゲーム盤が,玉で埋めつくされるとゲームオーバーとなります。玉を消すには玉を左クリックしてください。すると白い輪で玉がマークされ,移動可能になります。移動できる範囲は上下左右2マス以内で,斜めには移動できません。移動先にマウスカーソルを動かし,左クリックすると玉が移動します。玉の移動時に,同じ色の玉をまたぐよう移動すると,またがれた玉が消えます。マークのキャンセルは右クリックを押してください。

得点は玉をひとつ消すごとに10点。同じ色を続けて消すと,10,20,30点と点数が増えていきます。リプレイはリターンキー,終了はESCキーを押してください。

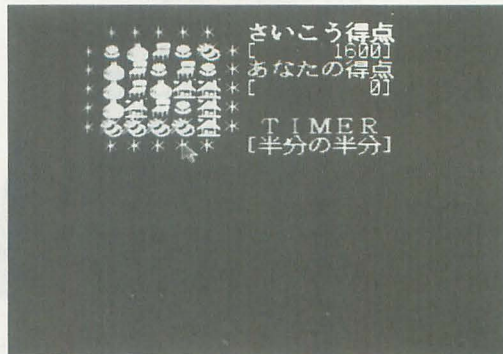
てことでこのSOLIXなんですが……。は,速い。そ,そうだ,私のマシンが16MHzだからいけないんだ! 10MHzに切り替えて……あう,まだ速い。

ただでさえ,パズルゲームっていうのは頭を使うのに,SOLIXは時間が制限されているのでまごまごしているうちにあっというまにゲームオーバーになってしまいますね。16MHzなど速いマシンで遊ぶと本当につらいっす。はっはっは。でも,テトリスやぷよぷよなんてのもそうでしたけど,アクション性のあるパズルって中毒性がありますよね。あとは対戦でもできるといいかもしれない,などと思ってしまいます。

難易度は,10MHzのX68000でコンパイルした場合を考えていますが,難易度テーブ

SCORE 360
LEVEL 3

SOLIX.BAS

さいこう得点
[1600]
あなたの得点
[0]
TIMER
[半分の半分]

MODOKI.C

ル（lvtbl）の数値を変えてもらえば，自分の好きな難易度にできます。かなりきついですが，調整すればBASICのままでも遊べないこともないかも……。やっぱりちょっとつらいかな？

コンパイラのある人はすぐコンパイルして，ない人はなんとか手に入れて遊んでみてくださいね。

時間がほしいの

それでは続いて２本目のプログラムにいきましょう。こいつもリアルタイムパズル（？）だったりします。MODOKI.Cです。どうぞ。

MODOKI.C for X680x0

（要Cコンパイラ）
茨城県　荒井一光

こちらはCコンパイラ用のプログラムですね。でも，リストを打ち込む前にちょっと下準備をしましょう。CONFIG.SYSに，

　USKCG＝¥USKCG.SYS

と書いておいてください。普通は書いてあるので大丈夫だと思いますが。次はSCORE.TXTという名前のテキストファイルをエディタなどで作り，その１行目から５行目までに「＊＊＊　0」という内容を書き込みセーブします。

これはハイスコアのテーブルで，ハイスコアが更新されるたびに，このファイルが書き換えられます。

こんどはエディタでMODOKI.Cという名前でリスト２を打ち込んで，セーブしてから，

　A>CC /Y /W MODOKI.C

とコンパイルすればOKです。

それではさっそく遊んでみましょう。

プログラムをスタートさせると画面左上に５種類25個のキャラクタがばらばらに表示されます。また，それを囲むように"＊"が表示されますね。この画面を見ればもうおわかりだと思いますが，それぞれ"＊"に対応する１行１列を上下左右にローテーションさせ，縦か横のどちらか一方に同じキャラクタを揃えていくというゲームです。ローテーションのやり方は"＊"を左クリックすることで，対応する行または列が動きます（例：上に動かすときは上の"＊"のいずれかを左クリック)。プレイ時間は約１分半。キャラクタを揃えるのにかかる時間によって得点が変化します。早く揃えて高得点を狙ってください。

このMODOKIはSOLIXと同じようにアクション性のあるパズルだけど，ちょっと趣が違います。こちらはSOLIXと比べるとなんとなく地味ですね。表示部分がテキスト色なのが原因なのかな。キャラクタごとに色を変えてみてもよかったかもしれないですね。あとは，展開が単調だから同時に何種類かのキャラクタを揃えたらボーナスが入ったり，残り時間が増えるといいかも。

なんとなくMODOKIのほうはルービックキューブの早解きに近いものがあるかな。SOLIXが攻めてくる敵ありのパズルだとすれば，こちらは自分との戦いって感じですね。テトリス以来最近は敵ありパズルっていうのが主流っぽくなってますけれども，どちらにもやっぱりそれぞれのゲーム性がありますよね，うん。得点がベスト５に入ると名前と得点か保存できるフィーチャーもちゃんとあるもんね。

高得点目指してレッツ　トライだ！

怖いけれど便利だ

さて，それでは今月の最後のプログラムですね。こいつはちょっと便利だけど怖いかもしれないディレクトリ移動ツール，DD.Xです。どうぞ。

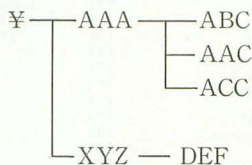
DD.X for X680x0

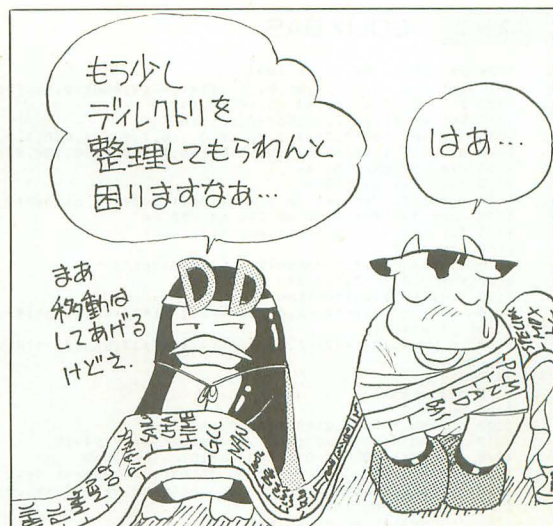
（要アセンブラ，リンカ）
富山県　高田靖之

このプログラムのリストはアセンブラのソースリストの形で掲載されています。このプログラムを実行するにはアセンブラとリンカが必要です。エディタでリスト３をDD.Sという名前で入力してから，それを保存します。アセンブラとリンカで（アセンブラがAS.X，リンカがLK.Xの場合)，

　A>AS DD.S
　A>LK DD.O

としてDD.Xを作ってください。

それではこのプログラムの使い方ですが，たとえば次のようなディレクトリ構造のディスクがあったとしましょう。

このプログラムは，

　A>DD　移りたいディレクトリの先頭から一部分

と入力すれば，自分がそのドライブのどのディレクトリにいてもそれに似た名前のディレクトリに移動してくれるというプログラムなのです。たとえば，

　A:¥XYZ¥DEF>DD AB

と入力すれば，A:¥AAA¥ABCに移動してくれるんですね。また，

　A:¥XYZ¥DEF>DD A

と入力したときは，A:¥AAAに移動します。ほかには，

　A:¥XYZ¥DEF>DD AA

と入力しても結果はA:¥AAAです。上のディレクトリ構造の場合，A:¥AAAの下のディレクトリにAACがありますが，A:¥AAAに移動します。

このディレクトリ移動の優先順位ですけれども，TREEコマンドなどで上のディレクトリ構造のように表示されたときにルート（¥）に近く，上にくるほうが優先となります。ルートに戻りたい場合は，

　A:¥XYZ¥DEF>DD ¥

で戻ります。

このコーナーにもよく投稿されるディレクトリ関係のツールです。このツールはちょっと便利そうだけど，ちょっと怖いプログラムですね。TRASHなんか選んでしまうと，どこに移るかわからなかったりして（わざわざTRASHに移るなよ)。

ちょっとわかりにくいですけど，自分でDD ○○なんてやってみて確かめてみてくださいね。私もやってみましたけど……。う～む，本当に予想外のところに移ってしまうんだな，これが。ただ単にディレクトリの整理整頓ができてないだけだったりして……。必要なのはディスクの整理……。

なんか聞きたくない話が出てしまったようなので，ここまでにしてしまいましょう。それでは，また来月！

リスト1 SOLIX.BAS

```
1000 /* SOLIX  By Shiro 1994.1
1010 int i,j,rx,ry,mx,my,bl,br,flg1,flg2,endflg,xx=100,yy=64
1020 int x1,y1,x2,y2,x3,y3,x4=20,y4=20
1025 int sc,lv,lvi,lvx,bo=10,cnt,tb
1030 dim int tbl(3,3)={ 1,0,1,0,0,1,0,1,1,0,1,0,0,1,0,1 }
1040 dim int lvtbl(9)={ 180,170,150,130,80,150,130,100,90,80 }
1050 dim str part(8),as
1060 dim char pat(255)
1070 m_init():for i=1 to 8:m_alloc(i,1000):m_assign(i,i):next
1080 part(1)="@54 T120 o5 l64 q8 v15 ce"
1090 for i=1 to 8:m_trk(i,part(i)):next
1100 sprite()
1110 screen 0,1,1,1:console ,,0:sp_disp(1)
1120 for j=0 to 3:for i=0 to 3
1130   if tbl(i,j)=0 then {
1140     fill(0+i*14+xx,0+j*16+yy,13+i*14+xx,15+j*16+yy,9)
1150   } else {
1160     fill(0+i*14+xx,0+j*16+yy,13+i*14+xx,15+j*16+yy,8):tbl(i
,j)=0
1170   }
1180 next:next
1190 while 1
1200   rx=0:ry=0:flg1=0:flg2=0:endflg=0
1210   x1=0:y1=0:x2=0:y2=0:x3=0:y3=0:x4=20:y4=20
1220   sc=0:lv=0:lvi=0:lvx=0:bo=10:cnt=0:tb=0
1230   for j=0 to 3:for i=0 to 3:tbl(i,j)=0:next:next
1240   cls:mouse(4):mouse(1):msarea(0+xx,0+yy,55+xx,63+yy)
1250   while endflg=0
1260     msstat(mx,my,bl,br)
1270     if br=-1 then { x4=20:y4=20:flg1=0 }
1280     if bl=0 then { flg2=0 }
1290     if flg1=0 and bl=-1 and flg2=0 then {set_rtn():flg2=1}
1300     if flg1=1 and bl=-1 and flg2=0 then {mov_rtn():flg2=1}
1310     sp_rnd():sp_hyouji()
1320   endwhile
1330   locate 12,13:print "GAMEOVER"
1340   while 1
1350     as=inkey$
1360     if as=chr$(&HD) then { break }
1370     if as=chr$(&H1B) then { end }
1380   endwhile
1390 endwhile
1400 func sprite()
1410   screen 0,2,1,1:sp_init()
1420   for i=1 to 3
1430     circle(7,7,6,i*2+1,,,260):paint(8,8,i*2+1)
1440     pset(4,6,15):pset(4,5,15):pset(5,4,15)
1450     get(0,0,15,15,pat):sp_def(i,pat)
1460   next
1470   fill(0,0,15,15,0):circle(7,7,7,15,,,260)
1480   get(0,0,15,15,pat):sp_def(4,pat)
1490 endfunc
1500 func sp_rnd()
1510   cnt=cnt+1
1520   if cnt<lvtbl(lvi) then { return() }
1530   cnt=0
1540   while 1
1550     rx=rnd()*4:ry=rnd()*4
1560     if tbl(rx,ry)=0 then { tbl(rx,ry)=rnd()*3+1:break }
1570   endwhile
1580   m_play(1)
1590 endfunc
1600 func sp_hyouji()
1610   locate 12,10:print "SCORE";sc
1620   locate 12,11:print "LEVEL";lv+1
1630   sp_move(0,x4*14+xx,y4*16+yy,4)
1640   k=0:endflg=1
1650   for j=0 to 3:for i=0 to 3
1660     k=k+1:sp_move(k,i*14+xx,j*16+yy,tbl(i,j))
1670     if tbl(i,j)=0 then { endflg=0 }
1680   next:next
1690 endfunc
1700 func set_rtn()
1710   mspos(mx,my)
1720   x1=(mx-xx)/14:y1=(my-yy)/16
1730   if tbl(x1,y1)>0 then { x4=x1:y4=y1:flg1=1 }
1740 endfunc
1750 func mov_rtn()
1760   mspos(mx,my)
1770   mx=(mx-xx)/14:my=(my-yy)/16
1780   x2=abs(x1-mx):y2=abs(y1-my)
1790   if tbl(mx,my)<>0 then { return() }
1800   if (x2=1 and y2=0) or (x2=0 and y2=1) then {
1810     tbl(mx,my)=tbl(x1,y1)
1820     tbl(x1,y1)=0
1830     x4=20:y4=20:flg1=0
1840     return()
1850   }
1860   if (x2=2 and y2=0) or (x2=0 and y2=2) then {
1870     x2=mx:y2=my:x3=mx:y3=my
1880     x3=x1+((x2-x1)/2):y3=y1+((y2-y1)/2)
1890     tbl(x2,y2)=tbl(x1,y1)
1900     tbl(x1,y1)=0
1910     if tbl(x2,y2)=tbl(x3,y3) then {
1920       if tbl(x3,y3)=tb then {
1930         bo=bo+10
1940       } else {
1950         bo=10
1960       }
1970       sc=sc+bo
1980       tb=tbl(x3,y3)
1990       tbl(x3,y3)=0
2000       lvx=lvx+1
2010       if lvx=10 then {
2020           lvx=0:lv=lv+1:lvi=lvi+1
2030           if lvi>9 then { lvi=9 }
2040       }
2050    }
2060    x4=20:y4=20:flg1=0
2070  }
2080 endfunc
```

リスト2 MODOKI.C

```
  1: #include <stdio.h>
  2: #include <basic0.h>
  3: #include <stdlib.h>
  4: #include <mouse.h>
  5: #include <iocslib.h>
  6: void move();  void print();  void problem();
  7: int choose(); int cheak();   void wait();
  8: #define WAKU  4     /* 枠(N*N)の時  WAKU=N-1 */
  9: #define TIMER 9000 /* TIMER 1分あたり 6000 ( ゲームの制限時間 ) */
 10: #define ZEN_F 0xEB /* 2バイト文字コード ( ZEN_F:1バイト目 ZEN_B:2
バイト目 ) */
 11: #define ZEN_B 0xAF /* WAKUを大きくするなら ZEN_B=0xA6 にするといいよ
*/
 12: int map[WAKU+1][WAKU+1];
 13: void main(void)
 14: {
 15:   struct{
 16:     char name[3];
 17:     int  score;
 18:   }keep[6],dumy1,dumy2;
 19:   int mx,my,bl,br,ct,houco,zahyo,time,first,tm_sc,tm,max;
 20:   FILE *fp;  fp=fopen("score.txt","r+");
 21:   b_init(); screen(1,1,1,1); b_csw(0);
 22:   mouse(4);  mouse(1); msarea(0,0,(WAKU+2)*16+16,(WAKU+3)*1
6);
 23:   do{
 24:     for(ct=0;ct<5;ct++)
 25:       fscanf(fp,"%3s %d",keep[ct].name,&keep[ct].score);
 26:     cls(); max=keep[0].score; keep[5].score=0;
 27:     for(ct=0;ct<WAKU+1;ct++){
 28:       locate(  ct*2+2,      0); printf("*");
 29:       locate(  ct*2+2,WAKU+2); printf("*");
 30:       locate(         0,  ct+1); printf("*");
 31:       locate(WAKU*2+4,  ct+1); printf("*");
 32:     }
 33:     color(7); locate(WAKU*2+6,0);        printf("さいこう得点");
 34:     color(3); locate(WAKU*2+6,1);        printf("[%10d]",max)
;
 35:     color(5); locate(WAKU*2+6,2);        printf("あなたの得点");
 36:     color(3); locate(WAKU*2+6,3);        printf("[%10d]",keep
[5].score);
 37:     color(6); locate(WAKU*2+6,WAKU+1); printf(" TIMER ");
 38:     color(3); locate(WAKU*2+6,WAKU+2); printf("[スタート!]");
 39:     first=ONTIME();
 40:     do{
 41:       problem(); print(0); tm_sc=ONTIME();
 42:       do{
 43:         time=ONTIME()-first; locate(WAKU*2+6,WAKU+2);
 44:         if(time>=TIMER/2    && time<TIMER/4    ){
 45:           color(7);printf("[あと半分!]");
 46:         }
 47:         if(time>=TIMER/4    && time<TIMER-1000){
 48:           color(5);printf("[半分の半分]");
 49:         }
 50:         if(time>=TIMER-1000 && time<TIMER      ){
 51:           color(6);printf("[もーちっと]");
 52:         }
 53:         if(choose(&houco,&zahyo)!=0){
 54:           move(houco,zahyo); color(3); print(0);
 55:           if(cheak()!=0){
 56:             tm=ONTIME()-tm_sc;
 57:             if(tm>=0   && tm<=200) ct=1000;
 58:             if(tm>=200 && tm<=480) ct=600;
 59:             if(tm>=480 && tm<=800) ct=300;
 60:             if(tm>800) ct=100;
 61:             keep[5].score=keep[5].score+ct;
 62:             if(keep[5].score>max) max=keep[5].score;
 63:             print(1); locate(2,3); printf("** %4d **",ct);
wait(100);
 64:             locate(WAKU*2+6,1); printf("[%10d]",max);
 65:             locate(WAKU*2+6,3); printf("[%10d]",keep[5].sco
re);
 66:             break;
 67:           }
 68:         }
 69:       }while(time<TIMER);
 70:     }while(time<TIMER);
 71:     locate(WAKU*2+6,WAKU+2); printf("[終り——!]"); print(1);
 72:     for(ct=0;ct<5;ct++){
 73:       if(keep[ct].score<=keep[5].score){
 74:         locate(2,1); printf("*記録更新*");
 75:         locate(2,3); printf(" 名前入力 ");
 76:         locate(2,4); printf("[          ]");
 77:         locate(6,4); scanf("%3s",keep[5].name);
 78:         dumy1=keep[ct]; keep[ct]=keep[5];
 79:         while(ct<5){
 80:           dumy2=keep[++ct];
 81:           keep[ct]=dumy1;
 82:           dumy1=dumy2;
 83:         }break;
 84:       }
 85:     }
```


▶コンセントが足りない。タコ足に流れる電流がオーバーしないか気がかりな今日この頃。
渡辺 久孝(27)大阪府

```
 86:      print(1); rewind(fp); color(5); locate(1,0); printf("*
BEST5*"); color(6);
 87:      for(ct=0;ct<5;ct++){
 88:        locate(1,ct+1);printf("%d. %3s %5d",ct+1,keep[ct].nam
e,keep[ct].score);
 89:        fprintf(fp,"%3s %d¥n",keep[ct].name,keep[ct].score);
 90:      }wait(300);
 91:      do{
 92:        color(9); locate(1,6); printf("続:左  終:右"),
 93:        msstat(&mx,&my,&bl,&br);
 94:      }while(bl==0 && br==0);
 95:      color(3);
 96:    }while(bl==-1);
 97:    fclose(fp); b_exit(0);
 98: }
 99: void move(int houco,int zahyo)
100: {
101:    int i,buf,xx,yy;
102:    switch(houco){
103:      case 1 : yy=0;    xx=zahyo; break;
104:      case 2 : yy=WAKU; xx=zahyo; break;
105:      case 3 : xx=0;    yy=zahyo; break;
106:      case 4 : xx=WAKU; yy=zahyo; break;
107:    }
108:    buf=map[xx][yy];
109:    if(houco<3)
110:      for(i=0;i<WAKU;i++)
111:        map[xx][abs(yy--)]=map[xx][abs(yy)];
112:    else
113:      for(i=0;i<WAKU;i++)
114:        map[abs(xx--)][yy]=map[abs(xx)][yy];
115:    map[abs(xx)][abs(yy)]=buf;
116: }
117: void print(int md)
118: {
119:    int x,y;
120:    for(y=0;y<WAKU+1;y++)
121:      for(x=0;x<WAKU+1;x++){
122:        if(md==0){ locate(x*2+2,y+1); printf("%c%c",ZEN_F,ZEN
_B+map[x][y]); }
123:        if(md==1){ locate(x*2+2,y+1); printf("  "); }
124:      }
125: }
126: void problem(void)
127: {
128:    int x,y,rd,count[WAKU+1];
129:    do{
130:      for(x=0;x<WAKU+1;x++) count[x]=WAKU+1;
131:      for(y=0;y<WAKU+1;y++){
132:        x=0;
133:        while(x<WAKU+1){
134:          rd=(int)((WAKU+1)*(rand()/32767.1));
135:          if(--count[rd]<0) continue;
136:          map[x++][y]=rd;
137:        }
138:      }
139:    }while(cheak()!=0);
140: }
141: int choose(int *houco,int *zahyo)
142: {
143:    int mx,my,bl,br,rt=0;
144:    msstat(&mx,&my,&bl,&br); mspos(&mx,&my);
145:    mx=(int)mx/16; my=(int)my/16;
146:    if(my==0      && bl==-1 && mx>0 && mx<WAKU+2){ *houco=1;
*zahyo=mx-1; rt=1; }
147:    if(my==WAKU+2 && bl==-1 && mx>0 && mx<WAKU+2){ *houco=2;
*zahyo=mx-1; rt=1; }
148:    if(mx==0      && bl==-1 && my>0 && my<WAKU+2){ *houco=3;
*zahyo=my-1; rt=1; }
149:    if(mx==WAKU+2 && bl==-1 && my>0 && my<WAKU+2){ *houco=4;
*zahyo=my-1; rt=1; }
150:    wait(8); return(rt);
151: }
152: int cheak(void)
153: {
154:    static int i,k,buf_x[WAKU+1],buf_y[WAKU+1];
155:    for(i=0;i<WAKU+1;i++){
156:      for(k=0;k<WAKU+1;k++){
157:        buf_x[map[k][i]]++;
158:        buf_y[map[i][k]]++;
159:      }
160:      for(k=0;k<WAKU+1;k++){
161:        if(buf_x[k]==WAKU+1 || buf_y[k]==WAKU+1) return(1);
162:        buf_x[k]=0; buf_y[k]=0;
163:      }
164:    }return(0);
165: }
166: void wait(int time)
167: {
168:    int tm=ONTIME();
169:    while(1)
170:      if(ONTIME()-tm>time) break;
171: }
```

リスト3 DD.S

```
  1:          INCLUDE IOCSCALL.MAC
  2:          INCLUDE DOSCALL.MAC
  3:
  4: PRINT    macro   R0
  5:          pea     R0
  6:          dc.w    _PRINT
  7:          addq.w  #4,sp
  8:          endm
  9:
 10:          .text
 11: * プログラムの開始点 ----------------------------------------
 12: START:   clr.w   d7
 13:          move.b  (a2)+,d7
 14:          dec.w   d7
 15:          bcs     ERR_1
 16:          lea     (a1),a4
 17:          lea     (a4),a5
 18:          move.b  #'¥',(a5)+
 19:          bsr     PLUS
 20:          add.w   #256,a1
 21:          clr.w   (a1)
 22:          cmp.b   #'¥',(a2)
 23:          beq     FIND2
 24:          cmp.b   #'.',(a2)
 25:          beq     FIND2
 26: * メインプログラム ------------------------------------------
 27:      * ディレクトリ切り替え後、一発目にはここへくる。
 28: LOOP_1: move.w   #%010000,-(sp)
 29:          pea     (a4)
 30:          pea     2(a1)
 31:          dc.w    _FILES
 32:          lea     10(sp),sp
 33:          btst    #31,d0
 34:          beq     JPOI_1
 35:      * ディレクトリが存在しませんでした。
 36: JPOI_2: tst.w    (a1)
 37:          beq     ERR_2
 38:          sub.w   #54+2,a1
 39:          dec.w   a5
 40:     LP2:cmp.b   #'¥',-(a5)
 41:          bne     LP2
 42:          inc.w   a5
 43:          clr.b   (a5)
 44:      * 同一ディレクトリを調べる場面。
 45: LOOP_2: pea      2(a1)
 46:          dc.w    _NFILES
 47:          addq.w  #4,sp
 48:          btst    #31,d0
 49:          bne     JPOI_2
 50: JPOI_1: * ファイル名が、規定のものかどうか調べます。
 51:          cmp.b   #'.',2+30(a1)
 52:          beq     LOOP_2
 53:          lea     2+30(a1),a0
 54:     LP3:move.b  (a0)+,(a5)+
 55:          bne     LP3
 56:          move.b  #'¥',-1(a5)
 57:          clr.b   (a5)
 58:          move.l  a2,a0
 59:          lea     2+30(a1),a3
 60:          move.w  d7,d0
 61:     LP1:cmp.b   (a3)+,(a0)+
 62:          dbne    d0,LP1
 63:          beq     FIND
 64:          add.w   #54+2,a1
 65:          move.w  #1,(a1)
 66:          bsr     PLUS
 67:          bra     LOOP_1
 68:
 69: * sub program -------------------------------------------
 70: FIND:    * 目的を果たしたので、正常終了します。
 71:          pea     (a4)
 72:          dc.w    _CHDIR
 73:          addq.w  #4,sp
 74:          dc.w    _EXIT
 75:
 76: FIND2:   * カレントを指定されたディレクトリに切り替えます。
 77:          pea     (a2)
 78:          dc.w    _CHDIR
 79:          addq.w  #4,sp
 80:          dc.w    _EXIT
 81:
 82: ERR_1:   * ディレクトリの指定がありませんでした。タイトルを表示して終了します。
 83:          print   TITLE(pc)
 84:          print   USAGE(pc)
 85:          move.w  #1,-(sp)
 86:          dc.w    _EXIT2
 87:
 88: ERR_2:   * 目的を果たせませんでした。終了します。
 89:          print   MES_1(pc)
 90:          move.w  #2,-(sp)
 91:          dc.w    _EXIT2
 92:
 93: PLUS:    * ファイル名を加工します。具体的には、ディレクトリにワイルドカードを付けて
います。
 94:          move.b  #'*',(a5)
 95:          move.b  #'.',1(a5)
 96:          move.b  #'*',2(a5)
 97:          clr.b   3(a5)
 98:          rts
 99:
100:          .data
101: * ワークエリア -------------------------------------------
102: TITLE:   .dc.b   'DD / カレントディレクトリ指定プログラム Version 1.1',
$D,$A
103:          .dc.b   '(C) Copyright 1993/11 by Y.T',$D,$A,0
104: USAGE:   .dc.b   ' USAGE:  DD  キーワード(ディレクトリ名の先頭より数文字
)',$D,$A
105:          .dc.b   ' ただし、DD ¥ は特例としてルートへ移動します。',$D,$A,0
106: MES_1:   .dc.b   'ディレクトリは見つかりませんでした。',$D,$A,0
107:          .even
108:
109:          .end    START
```

▶「卒業」のレビューで，なぜ教師がいかりや長介なのか気づくのに２日かかった。
高木　俊之(20)新潟県

SIDE A

いつかは鈴鹿

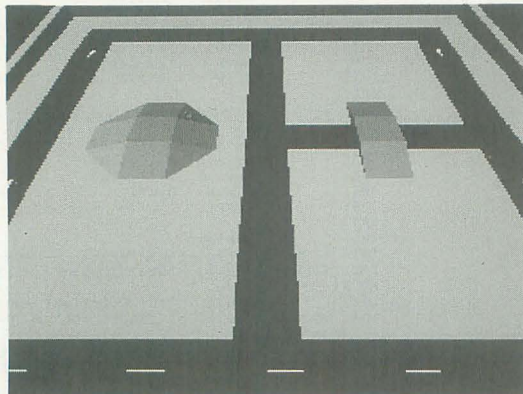
Tan Akihiko 丹 明彦

**今回は，いままで解説してきたアルゴリズムを使ったサンプルを紹介する
接触判定，斜面での旋回運動など，どのようにコーディングされたのか
リストを読みこなして理解を深めてもらいたい**

ハードルは高いが

さて，我々は世の中の規範となるべきドライビングシミュレータを制作するという崇高な目的のためにこの連載を続けている。そこでといってはなんだが，世間のレベルを知っておく必要性から，AMIGAやAT互換機向けの3DのF-1シミュレータである「WORLD CIRCUIT」で鈴鹿サーキットのタイムアタックに精を出している。というのは半分嘘で，単に面白いからはまっているだけである。

やり込めばやり込むほど「WORLD CIRCUIT」はすごいという認識を新たにさせられる。ドライビングシミュレータとして見てもカーレースゲームとして見ても2年前のものとは思えないレベルの高さだ。力学計算の確かさ，空間表現の精密さ，そしてたいして速くないマシンの上で立派に走らせてしまうプログラミングテクニック。

敵を知り己を知れば百戦危うからずという言葉があるが，走り込むほどに見えてくる「WORLD CIRCUIT」のレベルと，自分でぼちぼちと作り上げてきたプログラムのレベルを冷静に比べれば，戦うのなどやめて撤退するのが賢いのではないかと私の理性は警告している。

が，それでも私はわりと楽観的だ。あれこれと自分で作ってみて初めて見えてきたものもあるし，「WORLD CIRCUIT」にもわずかだが改善の余地がある。そのレベルに達することができるかどうかの見通しは決して明るくはないが，ま，やれるところまでやってみよう。

平坦なコースじゃつまらない

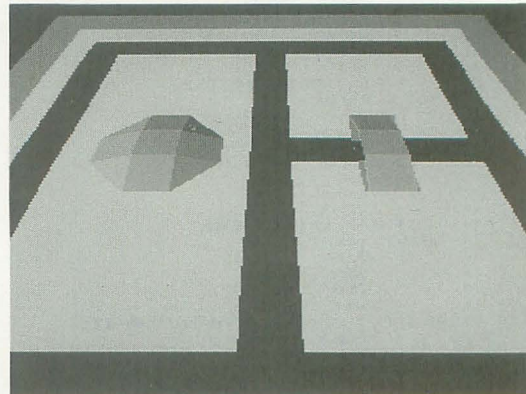
というわけで，今回は少しだけ複雑なテストコースを作り，その上で走り回るところまで作ることができた。当然ではあるが，本連載の過去の知識を動員している。テストコースには，まあ一般的な自動車レースを行うのに必要な要素を入れたつもりだ。できないのは「Hard Drivin'」くらいのものだろう。

具体的には，

1) 斜面がある
2) 壁がある
3) 立体交差がある

といったところ。これを先月号の能書きに従って実現してみた。

先月号の内容のうち，パラレルワールドは実装し

テストコース(表示用)

テストコース(接触判定用)

た。つまり，接触判定用の形状と表示用の形状の2つを用意している。多段階ディテールは実装しなかった。テストコースをすべてCのプログラムとして書いたのでしんどかったというのもあるが，まだコースが狭いのでそこまでする必要を感じなかった(復習しておくと，多段階ディテールは遠くの物体の形状を簡素化して表示速度を稼ぐための手法)。また今回はマップシステムもなし。これまたテストコースが狭いためと，SLASH ver.2.0が未公開のため採用を見送った。

さて，接触判定用の形状と表示用の形状の微妙な差に注意していただきたい。どちらの世界にも必要最小限の情報を記録している。表示用の形状は見てのとおり，せいぜい無駄なポリゴンを登録しないことくらいに留意しておけばいいだろう。接触判定用の形状はかなり簡略化されている。これは，過去に解説した接触判定用の構造体(checkinfo)にそのまま変換できる。それでは具体的な違いを見てみよう。まず道路のセンターラインがない。将来的には白線でタイヤが滑るということを表現することも考えられ，その場合はセンターラインも接触判定用の世界に出現するだろうが，現時点ではパス。それから壁もない。これは先月号で解説したとおり，壁と車との接触判定は壁そのものの形状では行わないためである。その役目を担うのは進入不可ゾーンである。壁の向こう側の床に敷いておき，これを踏んだら壁に衝突したとみなすものである。ただし，今回は進入不可ゾーンが実際の壁より少しはみ出している。その理由はあとで説明する。

プログラムの構造

今回のプログラムは1994年1月号で解説した円旋回運動プログラムの改良版である。前回は水平面での旋回運動だったが，今回は斜面もサポートした。

前回は意識しなかったことだが，メインループの構造は，

1) 操作
2) 運動
3) 姿勢算出

となっている。車はユーザーの操作によって速度や旋回半径を変化させる。次に，それに従って運動し新しい位置を得る。そして，その新しい位置にふさわしい姿勢を得る。最後に表示を行い，最初に戻るのである。

前回は特に3)の概念がなかった。というのは，常に水平面での運動ということが保証されていたため，つねに姿勢は水平であり，補正の必要もなかったというわけ。

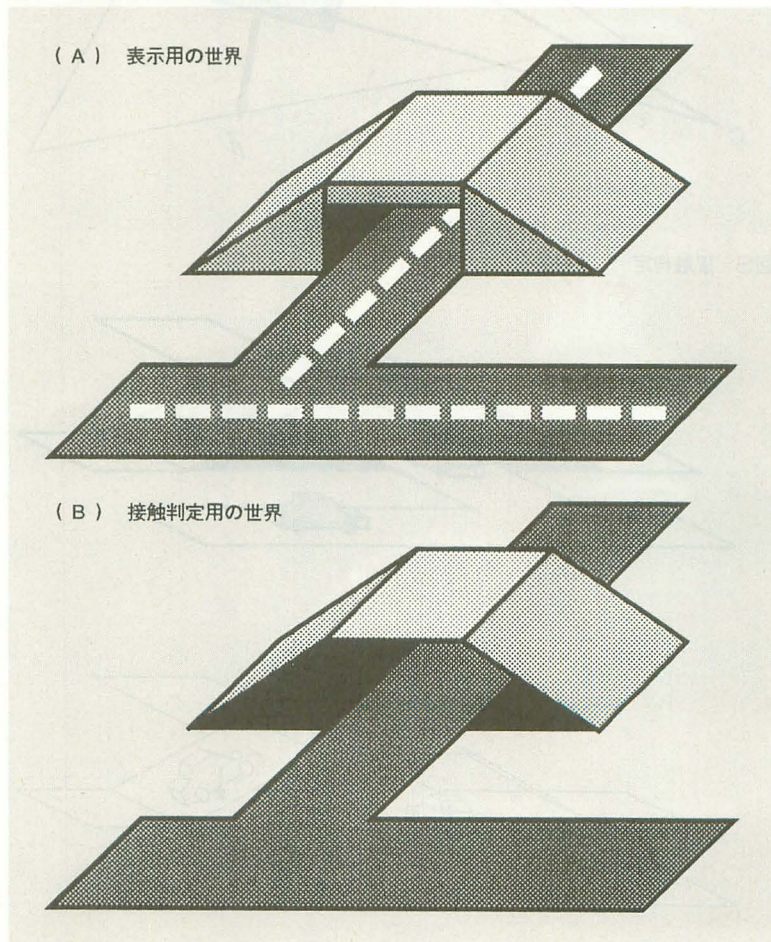
あと，細かいことだが，車がちょっと小さくなっている。前回はコースに比べて車が大きすぎ，自由自在に走り回れなかった。相対的にコースが大きくなればいいのだが，まだマップシステムを導入していないためコースを巨大化するわけにもいかず，逆に車を小さくしたのである。整数演算が多いので精度が多少心配だったのだが，あまり問題は出なかったようである。

斜面での旋回

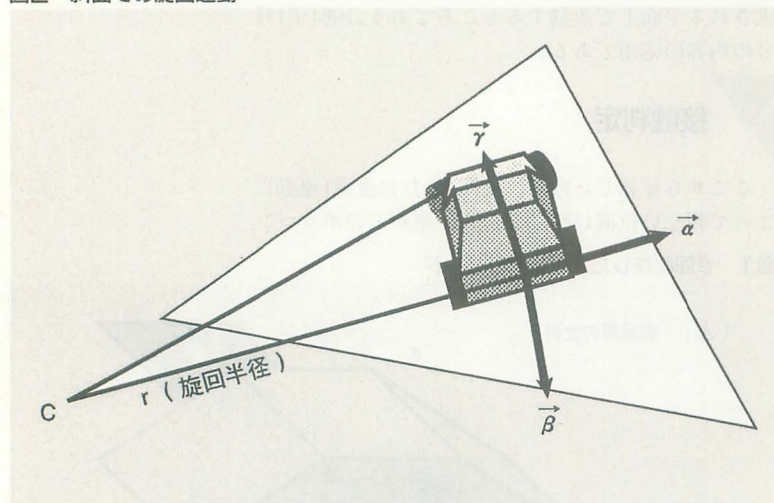
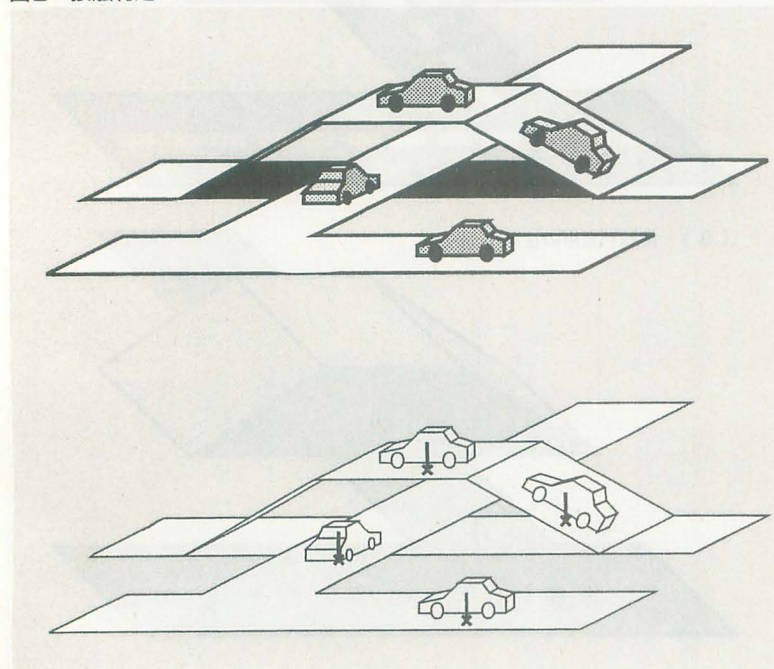
計算の基本的な部分はまったく前回の円旋回運動と同様である。異なるのは，前回が水平面すなわちzx平面上で運動していたのに対し，今回は車の基底座標系を基準とした水平面すなわちγ軸とα軸で構成される平面上で運動するところである。1994年1月号の内容の応用である。

接触判定

ここからが新しい内容。旋回(または直進)運動によって新しい位置に移動したら，車がどのポリゴン

図1　今回制作したパラレルワールド

を踏んでいるかを知る必要がある。斜面であれば姿勢を調節する必要があるし，進入不可ゾーンであれば車の動きを止める。そのための判定である。

とはいえ，やっているのは過去に解説したライブラリ(checklib)を利用することだけ。車の中央を検索起点として鉛直下向きに半直線を伸ばし，最初に接触したポリゴンがヒットポリゴンである。今回のバージョンは結構手抜きで，接触判定を車の中央の1点でしか行っていない。

車の踏むポリゴンを見つけるのであるから，車体の床面を検索起点にすればいいのではないかという意見もあろうが，それは少々危険なのだ。ひとつは計算誤差によって薄いポリゴンと交わっていないことになりかねないという懸念。もうひとつは（こちらがより重要なのだが)，いままで水平面を走っていて斜面に乗り始める瞬間に，車体の床面を検索起点にすると斜面のポリゴンがヒットしないためである。

さて，立体交差がなぜ可能かということについてひとこと解説をしておきたい。検索起点は車体の中央で，検索方向は鉛直下向き。ということは，立体交差の下を通るときは立体交差の下にあるポリゴンとしかヒットしないし，立体交差の上を通るときはまず立体交差の上にあるポリゴンとヒットする。こうして望みどおりの結果が得られる。

むろん，前提として接触判定ポリゴンリストの順序は表示用のそれに準じる必要がある。checklibの動作が，表示とは逆順に交差判定を行うように決められているためである(1994年2月号を参照)。表示して破綻のないように作れば交差判定もうまくいくし，計算量もそれほど多くない。結構バランスの取れた方法だと思うのであるが，いかがだろうか。

図2　斜面での旋回運動

図3　接触判定

壁との接触

これはつまり，進入不可ゾーンのポリゴンを踏んだかどうかという判定である。今回のプログラムでは，進入不可ゾーンのポリゴンをテストコースの周囲の壁や立体交差横の壁の向こう側にあたる部分の床に敷いてある。これらを踏んだかどうかの判定は，踏んでいるポリゴンの色で見ている。動けばいいという気分で作ったので，色ライブラリのポインタで比較するというあまりエレガントでない方法を用いている。

さて，コースデータの説明のところで，進入不可ゾーンのポリゴンが壁から少しはみ出すと書いたが，これは今回のプログラムが接触判定を車体の中央でしか行っていないためである。そのため車体サイズぶんだけ進入不可ゾーンを広く取ったのである。もしこれをやらないと，車はその中央が壁を越えるまで走るだろう。つまり車体が半分壁にめり込むまで衝突したとは判定されないのである。これを避けるために，とりあえず進入不可ゾーンを広げた。車体の四隅で接触判定するのが本筋であろうが，今回はこれでごまかした。

新しい姿勢の算出

これは，車があるポリゴンから傾きの変わる別のポリゴンに乗り移ったときに，車体が浮き上がったり沈み込んだりするのを防ぎ，常にポリゴン上に張りつくようにするためのものである。

車の位置は中心の座標(x,y,z)で，車の姿勢は基底座標系(α軸，β軸，γ軸)でそれぞれ表現される。こ

れを新しいポリゴンに沿わせる。具体的な計算方法は次のとおり。

1)　新しい位置に対応する高さを求める。これが新しいy座標になる。今回の実装ではx,z座標はいじらない

2)　ポリゴンの法線を求める。これが新しいβ軸になる

3)　γ軸の，新しいβ軸に垂直な成分を取り出す。これを正規化(長さを1にすること)したものが新しいγ軸になる

4)　新しいβ軸と新しいγ軸の外積を求める。これが新しいα軸になる

バグ

さて，以上の知識を用いれば，今回のテストコース上を走り回れるわけだが，今回のプログラムには実は恐ろしいバグがいる。ある条件下で車が地下に潜るのである！

原因は明確である。私自身，プログラムを書く前からこのバグは予想していた。めでたく（？）予想どおりになったというわけだ。車速を上げると，1フレームあたりの移動量が極端に大きくなる。このため，移動してから姿勢を補正するまでの暫定的な位置が新たに乗るべきポリゴンを突き破ってしまうことがあるのだ。斜面から水平面に降りた瞬間が特に危ない。

このバグはまだ対策が講じられていない。遅いマシンでは特に深刻。ほぼ確実に坂道の下で地下に潜るだろう。

対策はちょっと考えただけでも3つある。

1)　斜面の傾きの変化を小さくする(斜面ポリゴンを細かくする)ことで，バグの発生をある程度抑える。もちろん，極端に車速が上がった場合はどうしようもない

2)　タイムステップを小さくする。つまり，表示する周期よりも細かく接触判定を行う。これも，極端に車速が上がった場合はうまくいかない。なお，タイムステップを細かくする手法は，微妙なハンドル捌きやアクセルワークを実現する際に必須となる

3)　軌道とポリゴンの交点を求めて衝突地点を正確に調べる。これは計算コストが大きそうだが，上手にやればバグの発生を完璧に抑えられる

今後の展望

幾何学的な面での最後の難物は4輪の接地である。現時点では車体の中央だけを見て傾きを求めているので，なんとも動きが不自然だ。片輪が縁石に乗り上げるくらいはできたほうがいい。

そこから先は自動車力学の領域。サスペンション，路面との摩擦，操舵，トラクション，慣性モーメントなど，私自身も勉強せねばならないことが山のように控えている。これを越えれば4輪ドリフトもできるようになるだろう。きっと。

さて，幾度となく延期したが，今度こそ本当にSLASH ver.2.0をお見せできるだろう。行列と座標

図4　次の時刻の車体の姿勢の算出

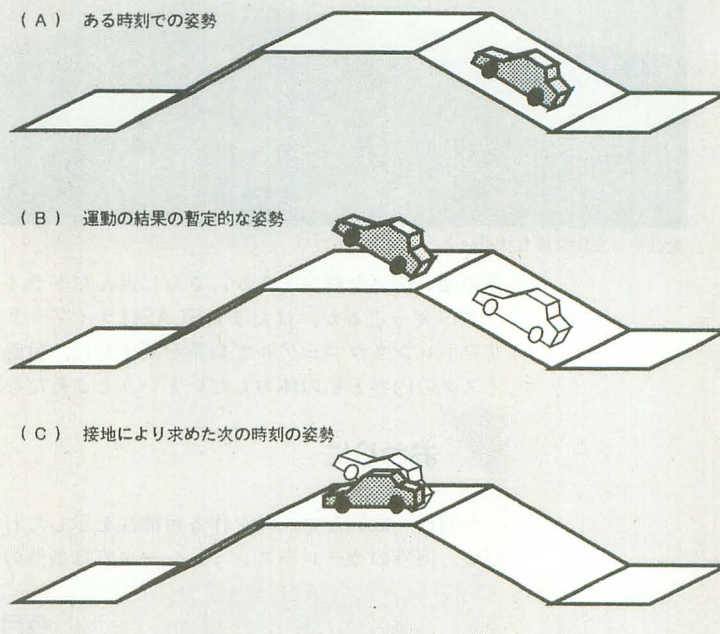
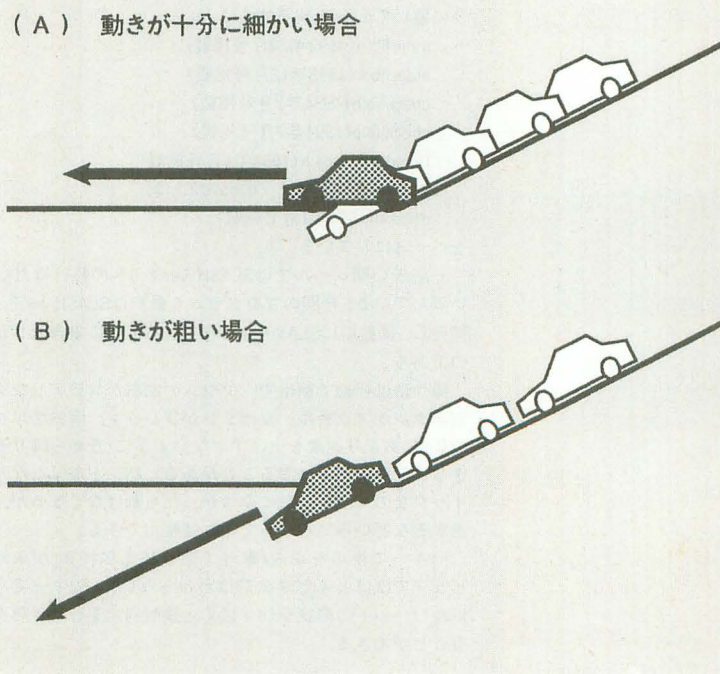
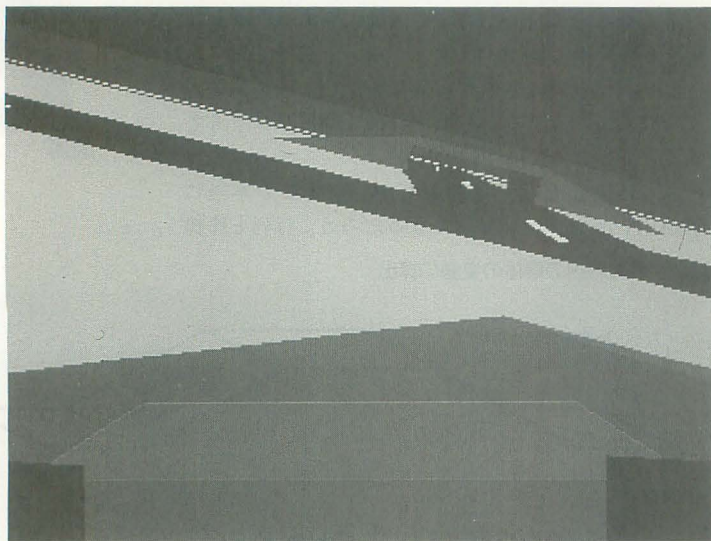



図5　車が地下に潜るバグ

ハードコア3Dエクスタシー(第7回)

走行中。立体交差もサポート

系の深い関係を解説するか，さらに進んだドライビングへ突っ走るか，はたまたSLASHライブラリのリファレンスマニュアルでお茶を濁すかは，付録ディスクの内容と私の体力しだいというところだな。

おわりに

今回は一般的なコースを作る可能性を示したわけだが，海外のカーレースシミュレータでは当然のように実現されているこれらの要素が，国産のカーレースゲームでは当然のように無視されているというのも妙な話だ。もっとリアリティをという意欲はまるで違う方面に向けられている気がしてならない。実在のF-1ドライバーやチームを登場させるためにライセンスを取るのはいい。大切なのは裏づけである。モータースポーツとしての深みである。せっかく大金を払って使わせてもらう実名も，肝心のレースがリアリティを与えるに足りない薄っぺらなものだと，リアルなはずの上っ面までも興醒めなものになってしまう。レーサーは走ってナンボの商売，優劣はサーキットで争うものであることを知るべきであろう。わけのわからないストーリーモードなどいらないのだ！

と思わず逆上してしまったが，別に私は現状を嘆いているわけではない。それほど人間がしおらしくできちゃいない。

ではどう思っているかって？　このぶんなら簡単に国内のトップクラスになれるんじゃないか，って思ってる。うーん尊大だな。

我々は最初に実在のデータを収集するというアプローチを取っていない。というよりもそういうアプローチが取れない。人手もないし，資金もないからね。だから地道に技術を積み重ねる。

というわけで，孤高を気取った連載は続くのである。それではまた次回。

今月のプログラム

とりあえずコンパイル方法。掲載リストを打ち込んだものと，以下のファイル(過去に掲載した)を同じディレクトリに置いてmakeを実行すること。

eulerlib.h(1993年12月号掲載)
eulerlib.c(1993年12月号掲載)
checklib.h(1994年2月号掲載)
checklib.c(1994年2月号掲載)
timedifference.h(1994年1月号掲載)

なお，メインルーチンのdtest2.cは，
dtest.c(1994年1月号掲載)
をベースにしている。

一応未公開レベルではSLASH ver.2.0への移行はだいたい済んでいる。今回のプログラムも最初はSLASH ver.2.0で開発し，掲載用にSLASH ver.1.0で通るように書き直したものである。

操作法は極めて簡単で，マウスの左右がステアリング，右ボタンがアクセル，左ボタンがブレーキ，両方のボタンで後退。あまり速度を上げすぎないように(丘から降りるときなどに車が地下に潜ることがある)。ハンドルはかなりクイックなので注意。壁にぶつかったら動けなくなるが，後退するなどいろいろやっていれば脱出できる。

F1キーで視点を車内/車外で切り替えられる(が車外のビューではほとんど車の形はわからない)。F2キーでパラレルワールドの形状を切り替え，接触判定を行う世界を見ることができる。

さて，車がユーザーの意志どおりに動かせるようになると，気になるのはレスポンスであろう。以下はdtest2.cにパフォーマンスモニタ(誌面の都合により掲載していない)処理を入れてコンパイルし，X68030によって測定した平均fps(frame per second，毎秒何コマ表示できるかという指標)値である。なお今回のサンプルでは数値演算コプロセッサは使用していない。

●SLASH ver.1
(eulerlibによって求めたオイラー角での指定)
……およそ8fps

●SLASH ver.2
(基底座標系から直接算出される行列での指定)
……およそ10fps

この程度の差でも結構体感できるから怖い。SLASH自体の高速化もさることながら，予想以上に重い実数計算の負荷がなくなったことが大きい。

速度的にはまあまあ希望がもてそう。実際の話，10fps出ていれば御の字ともいえるが，現在のプログラムは全画面クリアという間抜けをやっていたり，グラフィックVRAMしか使っていなかったりするため，高速化の余地はまだまだある。

はっきりいえば，私はX68030＋MC68882専用バージョンも作る気でいる。もちろん，それだけでは付録ディスクには載せられないので凝った手段を講じるつもりだ。乞うご期待。

SIDE

■リスト1　makefile

```
 1: SHELL = a:¥command.x
 2: SLASHLIBDIR = ..¥libl
 3: COLORDIR = ..¥color
 4: CCOPTS = -O -Wall
 5:
 6: %.o: %.c
 7:         gcc $(CCOPTS) -c $<
 8:
 9: all: dtest2.x
10:
11: dtest2.x: dtest2.o testcourse.o checklib.o eulerlib.o
12:         gcc -o dtest2.x testcourse.o dtest2.o checklib.o eulerlib.o¥
13:                 $(COLORDIR)¥tpllib.a¥
14:                 $(SLASHLIBDIR)¥_slashlib.a¥
15:                 $(SLASHLIBDIR)¥_utillib.a¥
16:                 $(SLASHLIBDIR)¥slashlib.a
17:
18: dtest2.o: dtest2.c eulerlib.h timedifference.h
19: testcourse.o: testcourse.c
20: checklib.o: checklib.c checklib.h
21: eulerlib.o: eulerlib.c eulerlib.h
22:
23: clean:
24:         if exist *.bak del -y *.bak
25:         if exist checklib.o del checklib.o
26:         if exist dtest2.o del dtest2.o
27:         if exist testcourse.o del testcourse.o
28:         if exist eulerlib.o del eulerlib.o
29:
30: distclean: clean
31:         if exist dtest2.x del dtest2.x
```

■リスト2　testcourse.c

```
  1: /*
  2:   testcourse.c
  3:   - テストコース
  4:     Feb. 1994 丹 明彦(Oh!X)
  5: */
  6:
  7: #include  <stdlib.h>
  8: #include  "..¥lib¥_slashlib.h"
  9: #include  "..¥color¥_tpllib.h"
 10:
 11: SLPOLYGONLIST *field_polygonlist;
 12: SLPOINTLIST   *field_pointlist;
 13:
 14: SLPOLYGONLIST *collide_polygonlist;
 15: SLPOINTLIST   *collide_pointlist;
 16:
 17: #define COLLIDE_TOLERANCE 25
 18:
 19: /* フィールド */
 20: void  create_field()
 21: {
 22:   int x, z, i, j;
 23:
 24: #define FS  2048        /* フィールドのサイズ */
 25: #define FD  8           /* フィールドの分割数 */
 26: #define RW  (FS/16)     /* 道路の幅 */
 27: #define LW  (FS/512)    /* センターラインの幅 */
 28: #define LN  16          /* センターラインの数 */
 29: #define LL  (FS/LN)     /* センターラインの長さ */
 30: #define WH  128         /* 壁の高さ */
 31: #define HH  64          /* 丘などの高さ */
 32: #define HW  (FS/16)     /* 丘などの幅 */
 33:
 34:   field_polygonlist = malloc( sizeof(SLPOLYGONLIST)+sizeof(SLPOLYG
ON)*200 );
 35:   field_pointlist = malloc( sizeof(SLPOINTLIST)+sizeof(SLPOINT)*80
0 );
 36:   field_polygonlist->n = 0;
 37:   field_pointlist->n = 0;
 38:
 39:   collide_polygonlist = malloc( sizeof(SLPOLYGONLIST)+sizeof(SLPOL
YGON)*200 );
 40:   collide_pointlist = malloc( sizeof(SLPOINTLIST)+sizeof(SLPOINT)*
800 );
 41:   collide_polygonlist->n = 0;
 42:   collide_pointlist->n = 0;
 43:
 44:   /* 床 */
 45:   for ( i = 0; i < FD; i++ ) {
 46:     x = -FS + i*(FS*2/FD);
 47:     for ( j = 0; j < FD; j++ ) {
 48:       z = -FS + j*(FS*2/FD);
 49:       addtetragon( field_polygonlist, field_pointlist,
 50:         x,            0, z,
 51:         x,            0, z+(FS*2/FD),
 52:         x+(FS*2/FD), 0, z+(FS*2/FD),
 53:         x+(FS*2/FD), 0, z,
 54:         &std_yellow );
 55:       addtetragon( collide_polygonlist, collide_pointlist,
 56:         x,            0, z,
 57:         x,            0, z+(FS*2/FD),
 58:         x+(FS*2/FD), 0, z+(FS*2/FD),
 59:         x+(FS*2/FD), 0, z,
 60:         &std_yellow );
 61:     }
 62:   }
 63:
 64:   /* 壁 */
 65:   x = -FS+(FS/FD);
 66:   for ( j = 0; j < FD-1; j++ ) {
 67:     z = -FS + j*(FS*2/FD) + (FS/FD);
 68:     addtetragon( field_polygonlist, field_pointlist,
 69:       x,   0, z,
 70:       x, -WH, z,
 71:       x, -WH, z+(FS*2/FD),
 72:       x,   0, z+(FS*2/FD),
 73:       &std_lightgray );
 74:   }
 75:   x = FS-(FS/FD);
 76:   for ( j = 0; j < FD-1; j++ ) {
 77:     z = -FS + j*(FS*2/FD) + (FS/FD);
 78:     addtetragon( field_polygonlist, field_pointlist,
 79:       x, -WH, z,
 80:       x,   0, z,
 81:       x,   0, z+(FS*2/FD),
 82:       x, -WH, z+(FS*2/FD),
 83:       &std_lightgray );
 84:   }
 85:   z = -FS+(FS/FD);
 86:   for ( i = 0; i < FD-1; i++ ) {
 87:     x = -FS + i*(FS*2/FD) + (FS/FD);
 88:     addtetragon( field_polygonlist, field_pointlist,
 89:       x,              -WH, z,
 90:       x,                0, z,
 91:       x+(FS*2/FD),      0, z,
 92:       x+(FS*2/FD), -WH, z,
 93:       &std_lightgray );
 94:   }
 95:   z = FS-(FS/FD);
 96:   /*for ( i = 0; i < FD; i++ ) {*/
 97:   for ( i = 0; i < FD-1; i++ ) {
 98:     /*x = -FS + i*(FS*2/FD);*/
 99:     x = -FS + i*(FS*2/FD) + (FS/FD);  /* 半ずらし */
100:     addtetragon( field_polygonlist, field_pointlist,
101:       x,                0, z,
102:       x,              -WH, z,
103:       x+(FS*2/FD), -WH, z,
104:       x+(FS*2/FD),      0, z,
105:       &std_lightgray );
106:   }
107:
108:   /* 場外 */
109:   x = -FS;
110:   for ( j = 0; j < FD; j++ ) {
111:     z = -FS + j*(FS*2/FD);
112:     addtetragon( collide_polygonlist, collide_pointlist,
113:       x-COLLIDE_TOLERANCE,           0, z-COLLIDE_TOLERANCE,
114:       x-COLLIDE_TOLERANCE,           0, z+(FS*2/FD)+COLLIDE_TOLERANC
E,
115:       x+(FS/FD)+COLLIDE_TOLERANCE, 0, z+(FS*2/FD)+COLLIDE_TOLERANC
E,
116:       x+(FS/FD)+COLLIDE_TOLERANCE, 0, z-COLLIDE_TOLERANCE,
117:       &std_magenta );
118:   }
119:   x = FS-(FS/FD);
120:   for ( j = 0; j < FD; j++ ) {
121:     z = -FS + j*(FS*2/FD);
122:     addtetragon( collide_polygonlist, collide_pointlist,
123:       x-COLLIDE_TOLERANCE,           0, z-COLLIDE_TOLERANCE,
124:       x-COLLIDE_TOLERANCE,           0, z+(FS*2/FD)+COLLIDE_TOLERANC
E,
125:       x+(FS/FD)+COLLIDE_TOLERANCE, 0, z+(FS*2/FD)+COLLIDE_TOLERANC
E,
126:       x+(FS/FD)+COLLIDE_TOLERANCE, 0, z-COLLIDE_TOLERANCE,
127:       &std_magenta );
128:   }
129:   z = -FS;
130:   for ( i = 0; i < FD; i++ ) {
131:     x = -FS + i*(FS*2/FD);
132:     addtetragon( collide_polygonlist, collide_pointlist,
133:       x-COLLIDE_TOLERANCE,             0, z-COLLIDE_TOLERANCE,
134:       x-COLLIDE_TOLERANCE,             0, z+(FS/FD)+COLLIDE_TOLERANC
E,
135:       x+(FS*2/FD)+COLLIDE_TOLERANCE, 0, z+(FS/FD)+COLLIDE_TOLERANC
E,
136:       x+(FS*2/FD)+COLLIDE_TOLERANCE, 0, z-COLLIDE_TOLERANCE,
137:       &std_magenta );
138:   }
139:   z = FS-(FS/FD);  /* ブロックサイズの半分 */
140:   for ( i = 0; i < FD; i++ ) {
141:     x = -FS + i*(FS*2/FD);
142:     addtetragon( collide_polygonlist, collide_pointlist,
143:       x-COLLIDE_TOLERANCE,             0, z-COLLIDE_TOLERANCE,
144:       x-COLLIDE_TOLERANCE,             0, z+(FS/FD)+COLLIDE_TOLERANC
E,
145:       x+(FS*2/FD)+COLLIDE_TOLERANCE, 0, z+(FS/FD)+COLLIDE_TOLERANC
E,
146:       x+(FS*2/FD)+COLLIDE_TOLERANCE, 0, z-COLLIDE_TOLERANCE,
147:       &std_magenta );
148:   }
149:
150:   /* 道路 */
151:   x = -FS+(FS*2/FD)+RW;
152:   for ( j = 1; j < FD-1; j++ ) {
153:     z = -FS + j*(FS*2/FD);
154:     addtetragon( field_polygonlist, field_pointlist,
155:       x-RW, 0, z,
156:       x-RW, 0, z+(FS*2/FD),
157:       x+RW, 0, z+(FS*2/FD),
158:       x+RW, 0, z,
```

```
159:       &std_darkblue );
160:     addtetragon( field_polygonlist, field_pointlist,
161:       x-LW, 0, z,
162:       x-LW, 0, z+LL,
163:       x+LW, 0, z+LL,
164:       x+LW, 0, z,
165:       &std_white );
166:     addtetragon( collide_polygonlist, collide_pointlist,
167:       x-RW, 0, z,
168:       x-RW, 0, z+(FS*2/FD),
169:       x+RW, 0, z+(FS*2/FD),
170:       x+RW, 0, z,
171:       &std_darkblue );
172:   }
173:   x = 0;
174:   for ( j = 1; j < FD-1; j++ ) {
175:     z = -FS + j*(FS*2/FD);
176:     addtetragon( field_polygonlist, field_pointlist,
177:       x-RW, 0, z,
178:       x-RW, 0, z+(FS*2/FD),
179:       x+RW, 0, z+(FS*2/FD),
180:       x+RW, 0, z,
181:       &std_darkblue );
182:     addtetragon( field_polygonlist, field_pointlist,
183:       x-LW, 0, z,
184:       x-LW, 0, z+LL,
185:       x+LW, 0, z+LL,
186:       x+LW, 0, z,
187:       &std_white );
188:     addtetragon( collide_polygonlist, collide_pointlist,
189:       x-RW, 0, z,
190:       x-RW, 0, z+(FS*2/FD),
191:       x+RW, 0, z+(FS*2/FD),
192:       x+RW, 0, z,
193:       &std_darkblue );
194:   }
195:   x = FS-(FS*2/FD)-RW;
196:   for ( j = 1; j < FD-1; j++ ) {
197:     z = -FS + j*(FS*2/FD);
198:     addtetragon( field_polygonlist, field_pointlist,
199:       x-RW, 0, z,
200:       x-RW, 0, z+(FS*2/FD),
201:       x+RW, 0, z+(FS*2/FD),
202:       x+RW, 0, z,
203:       &std_darkblue );
204:     addtetragon( field_polygonlist, field_pointlist,
205:       x-LW, 0, z,
206:       x-LW, 0, z+LL,
207:       x+LW, 0, z+LL,
208:       x+LW, 0, z,
209:       &std_white );
210:     addtetragon( collide_polygonlist, collide_pointlist,
211:       x-RW, 0, z,
212:       x-RW, 0, z+(FS*2/FD),
213:       x+RW, 0, z+(FS*2/FD),
214:       x+RW, 0, z,
215:       &std_darkblue );
216:   }
217:   z = -FS+(FS*2/FD)+RW;
218:   for ( i = 1; i < FD-1; i++ ) {
219:     x = -FS + i*(FS*2/FD);
220:     addtetragon( field_polygonlist, field_pointlist,
221:       x,            0, z-RW,
222:       x,            0, z+RW,
223:       x+(FS*2/FD), 0, z+RW,
224:       x+(FS*2/FD), 0, z-RW,
225:       &std_darkblue );
226:     addtetragon( field_polygonlist, field_pointlist,
227:       x    , 0, z-LW,
228:       x    , 0, z+LW,
229:       x+LL, 0, z+LW,
230:       x+LL, 0, z-LW,
231:       &std_white );
232:     addtetragon( collide_polygonlist, collide_pointlist,
233:       x,            0, z-RW,
234:       x,            0, z+RW,
235:       x+(FS*2/FD), 0, z+RW,
236:       x+(FS*2/FD), 0, z-RW,
237:       &std_darkblue );
238:   }
239:   z = 0;
240:   for ( i = (FD/2); i < FD-1; i++ ) {
241:     x = -FS + i*(FS*2/FD);
242:     addtetragon( field_polygonlist, field_pointlist,
243:       x,            0, z-RW,
244:       x,            0, z+RW,
245:       x+(FS*2/FD), 0, z+RW,
246:       x+(FS*2/FD), 0, z-RW,
247:       &std_darkblue );
248:     addtetragon( field_polygonlist, field_pointlist,
249:       x    , 0, z-LW,
250:       x    , 0, z+LW,
251:       x+LL, 0, z+LW,
252:       x+LL, 0, z-LW,
253:       &std_white );
254:     addtetragon( collide_polygonlist, collide_pointlist,
255:       x,            0, z-RW,
256:       x,            0, z+RW,
257:       x+(FS*2/FD), 0, z+RW,
258:       x+(FS*2/FD), 0, z-RW,
259:       &std_darkblue );
260:   }
261:   z = FS-(FS*2/FD)-RW;
262:   for ( i = 1; i < FD-1; i++ ) {
263:     x = -FS + i*(FS*2/FD);
264:     addtetragon( field_polygonlist, field_pointlist,
265:       x,            0, z-RW,
266:       x,            0, z+RW,
267:       x+(FS*2/FD), 0, z+RW,
268:       x+(FS*2/FD), 0, z-RW,
269:       &std_darkblue );
270:     addtetragon( field_polygonlist, field_pointlist,
271:       x    , 0, z-LW,
272:       x    , 0, z+LW,
273:       x+LL, 0, z+LW,
274:       x+LL, 0, z-LW,
275:       &std_white );
276:     addtetragon( collide_polygonlist, collide_pointlist,
277:       x,            0, z-RW,
278:       x,            0, z+RW,
279:       x+(FS*2/FD), 0, z+RW,
280:       x+(FS*2/FD), 0, z-RW,
281:       &std_darkblue );
282:   }
283:
284:   /* トンネルと丘 */
285:   x = -FS/3;
286:   z = 0;
287:
288:   /* トンネル(内)……fieldのみ */
289:   addtetragon( field_polygonlist, field_pointlist,
290:     -x-HW,          0, z+HW,
291:     -x-HW, -HH*3/4, z+HW,
292:     -x+HW, -HH*3/4, z+HW,
293:     -x+HW,          0, z+HW,
294:     &std_green );
295:   addtetragon( field_polygonlist, field_pointlist,
296:     -x+HW, -HH*3/4, z+HW,
297:     -x-HW, -HH*3/4, z+HW,
298:     -x-HW, -HH*3/4, z-HW,
299:     -x+HW, -HH*3/4, z-HW,
300:     &std_green );
301:   addtetragon( field_polygonlist, field_pointlist,
302:     -x-HW, -HH*3/4, z-HW,
303:     -x-HW,          0, z-HW,
304:     -x+HW,          0, z-HW,
305:     -x+HW, -HH*3/4, z-HW,
306:     &std_green );
307:
308:   /* トンネル(底)……collideのみ */
309:   addtetragon( collide_polygonlist, collide_pointlist,
310:     -x-HW-COLLIDE_TOLERANCE, 0, z+HW-COLLIDE_TOLERANCE,
311:     -x-HW-COLLIDE_TOLERANCE, 0, z+HW*3/*+COLLIDE_TOLERANCE*/,
312:     -x+HW+COLLIDE_TOLERANCE, 0, z+HW*3/*+COLLIDE_TOLERANCE*/,
313:     -x+HW+COLLIDE_TOLERANCE, 0, z+HW-COLLIDE_TOLERANCE,
314:     &std_magenta );
315:   addtetragon( collide_polygonlist, collide_pointlist,
316:     -x-HW-COLLIDE_TOLERANCE, 0, z-HW*3/*-COLLIDE_TOLERANCE*/,
317:     -x-HW-COLLIDE_TOLERANCE, 0, z-HW+COLLIDE_TOLERANCE,
318:     -x+HW+COLLIDE_TOLERANCE, 0, z-HW+COLLIDE_TOLERANCE,
319:     -x+HW+COLLIDE_TOLERANCE, 0, z-HW*3/*-COLLIDE_TOLERANCE*/,
320:     &std_magenta );
321:
322:   /* 丘(右) */
323:   addtetragon( field_polygonlist, field_pointlist,
324:     x+HW  , -HH, z+HW,
325:     x+HW*3,   0, z+HW,
326:     x+HW*3,   0, z-HW,
327:     x+HW  , -HH, z-HW,
328:     &std_green );
329:   addtriangle( field_polygonlist, field_pointlist,
330:     x+HW  , -HH, z+HW,
331:     x+HW  ,   0, z+HW*3,
332:     x+HW*3,   0, z+HW,
333:     &std_green );
334:   addtriangle( field_polygonlist, field_pointlist,
335:     x+HW  , -HH, z-HW,
336:     x+HW*3,   0, z-HW,
337:     x+HW  ,   0, z-HW*3,
338:     &std_green );
339:   addtetragon( collide_polygonlist, collide_pointlist,
340:     x+HW  , -HH, z+HW,
341:     x+HW*3,   0, z+HW,
342:     x+HW*3,   0, z-HW,
343:     x+HW  , -HH, z-HW,
344:     &std_green );
345:   addtriangle( collide_polygonlist, collide_pointlist,
346:     x+HW  , -HH, z+HW,
347:     x+HW  ,   0, z+HW*3,
348:     x+HW*3,   0, z+HW,
349:     &std_green );
350:   addtriangle( collide_polygonlist, collide_pointlist,
351:     x+HW  , -HH, z-HW,
352:     x+HW*3,   0, z-HW,
353:     x+HW  ,   0, z-HW*3,
354:     &std_green );
355:
356:   /* トンネル(左)……fieldのみ */
357:   addtetragon( field_polygonlist, field_pointlist,
358:     -x-HW, -HH*3/4, z+HW,
359:     -x-HW, -HH    , z+HW,
360:     -x-HW, -HH    , z-HW,
361:     -x-HW, -HH*3/4, z-HW,
362:     &std_green );
363:   addtriangle( field_polygonlist, field_pointlist,
364:     -x-HW,   0, z+HW*3,
365:     -x-HW, -HH, z+HW,
366:     -x-HW,   0, z+HW,
367:     &std_green );
368:   addtriangle( field_polygonlist, field_pointlist,
369:     -x-HW, -HH, z-HW,
370:     -x-HW,   0, z-HW*3,
371:     -x-HW,   0, z-HW,
372:     &std_green );
373:
374:   /* 丘(中) */
375:   addtetragon( field_polygonlist, field_pointlist,
376:     x-HW, -HH, z+HW,
377:     x-HW,   0, z+HW*3,
378:     x+HW,   0, z+HW*3,
379:     x+HW, -HH, z+HW,
380:     &std_green );
```

▶「おもちゃのカンヅメ」が当たる前に糖尿病で死ぬかもしれない……。
大久保　明弘(21)岩手県

```
381:   addtetragon( field_polygonlist, field_pointlist,
382:     x-HW, -HH, z+HW,
383:     x+HW, -HH, z+HW,
384:     x+HW, -HH, z-HW,
385:     x-HW, -HH, z-HW,
386:     &std_green );
387:   addtetragon( field_polygonlist, field_pointlist,
388:     x-HW,   0, z-HW*3,
389:     x-HW, -HH, z-HW,
390:     x+HW, -HH, z-HW,
391:     x+HW,   0, z-HW*3,
392:     &std_green );
393:   addtetragon( collide_polygonlist, collide_pointlist,
394:     x-HW, -HH, z+HW,
395:     x-HW,   0, z+HW*3,
396:     x+HW,   0, z+HW*3,
397:     x+HW, -HH, z+HW,
398:     &std_green );
399:   addtetragon( collide_polygonlist, collide_pointlist,
400:     x-HW, -HH, z+HW,
401:     x+HW, -HH, z+HW,
402:     x+HW, -HH, z-HW,
403:     x-HW, -HH, z-HW,
404:     &std_green );
405:   addtetragon( collide_polygonlist, collide_pointlist,
406:     x-HW,   0, z-HW*3,
407:     x-HW, -HH, z-HW,
408:     x+HW, -HH, z-HW,
409:     x+HW,   0, z-HW*3,
410:     &std_green );
411:
412:   /* トンネル(上) */
413:   addtetragon( field_polygonlist, field_pointlist,
414:     -x-HW, -HH, z+HW,
415:     -x-HW,   0, z+HW*3,
416:     -x+HW,   0, z+HW*3,
417:     -x+HW, -HH, z+HW,
418:     &std_green );
419:   addtetragon( field_polygonlist, field_pointlist,
420:     -x-HW, -HH, z+HW,
421:     -x+HW, -HH, z+HW,
422:     -x+HW, -HH, z-HW,
423:     -x-HW, -HH, z-HW,
424:     &std_green );
425:   addtetragon( field_polygonlist, field_pointlist,
426:     -x-HW,   0, z-HW*3,
427:     -x-HW, -HH, z-HW,
428:     -x+HW, -HH, z-HW,
429:     -x+HW,   0, z-HW*3,
430:     &std_green );
431:   addtetragon( collide_polygonlist, collide_pointlist,
432:     -x-HW, -HH, z+HW,
433:     -x-HW,   0, z+HW*3,
434:     -x+HW,   0, z+HW*3,
435:     -x+HW, -HH, z+HW,
436:     &std_green );
437:   addtetragon( collide_polygonlist, collide_pointlist,
438:     -x-HW, -HH, z+HW,
439:     -x+HW, -HH, z+HW,
440:     -x+HW, -HH, z-HW,
441:     -x-HW, -HH, z-HW,
442:     &std_green );
443:   addtetragon( collide_polygonlist, collide_pointlist,
444:     -x-HW,   0, z-HW*3,
445:     -x-HW, -HH, z-HW,
446:     -x+HW, -HH, z-HW,
447:     -x+HW,   0, z-HW*3,
448:     &std_green );
449:
450:   /* 丘(左) */
451:   addtetragon( field_polygonlist, field_pointlist,
452:     x-HW*3,   0, z+HW,
453:     x-HW  , -HH, z+HW,
454:     x-HW  , -HH, z-HW,
455:     x-HW*3,   0, z-HW,
456:     &std_green );
457:   addtriangle( field_polygonlist, field_pointlist,
458:     x-HW  ,   0, z+HW*3,
459:     x-HW  , -HH, z+HW,
460:     x-HW*3,   0, z+HW,
461:     &std_green );
462:   addtriangle( field_polygonlist, field_pointlist,
463:     x-HW*3,   0, z-HW,
464:     x-HW  , -HH, z-HW,
465:     x-HW  ,   0, z-HW*3,
466:     &std_green );
467:   addtetragon( collide_polygonlist, collide_pointlist,
468:     x-HW*3,   0, z+HW,
469:     x-HW  , -HH, z+HW,
470:     x-HW  , -HH, z-HW,
471:     x-HW*3,   0, z-HW,
472:     &std_green );
473:   addtriangle( collide_polygonlist, collide_pointlist,
474:     x-HW  ,   0, z+HW*3,
475:     x-HW  , -HH, z+HW,
476:     x-HW*3,   0, z+HW,
477:     &std_green );
478:   addtriangle( collide_polygonlist, collide_pointlist,
479:     x-HW*3,   0, z-HW,
480:     x-HW  , -HH, z-HW,
481:     x-HW  ,   0, z-HW*3,
482:     &std_green );
483:
484:   /* トンネル(右)……fieldのみ */
485:   addtetragon( field_polygonlist, field_pointlist,
486:     -x+HW, -HH     , z+HW,
487:     -x+HW, -HH*3/4, z+HW,
488:     -x+HW, -HH*3/4, z-HW,
489:     -x+HW, -HH     , z-HW,
490:     &std_green );
491:   addtriangle( field_polygonlist, field_pointlist,
492:     -x+HW, -HH, z+HW,
493:     -x+HW,   0, z+HW*3,
494:     -x+HW,   0, z+HW,
495:     &std_green );
496:   addtriangle( field_polygonlist, field_pointlist,
497:     -x+HW,   0, z-HW*3,
498:     -x+HW, -HH, z-HW,
499:     -x+HW,   0, z-HW,
500:     &std_green );
501:
502:   AddNorm( field_polygonlist, field_pointlist );
503:   AddNorm( collide_polygonlist, collide_pointlist );
504:
505:   return;
506: }
507:
508: void  destroy_field()
509: {
510:   free( field_polygonlist );
511:   free( field_pointlist );
512:   free( collide_polygonlist );
513:   free( collide_pointlist );
514:   return;
515: }
```

■リスト3 dtest2.c

```
 1: /*
 2:   dtest2.c
 3:   - 車の動作(発展型)
 4:     Feb. 1994 丹 明彦(Oh!X)
 5: */
 6:
 7: #define   __IOCS_INLINE__
 8: #include  <iocslib.h>
 9: #define   __DOS_INLINE__
10: #include  <doslib.h>
11: #include  <stdio.h>
12: #include  <stdlib.h>
13:
14: #include  "..¥lib¥_slashlib.h"
15: #include  "..¥color¥_tpllib.h"
16: #include  "eulerlib.h"
17: #include  "timedifference.h"
18: #include  "checklib.h"
19:
20: #define N_POINT   800
21: #define N_OBJECT  4
22:
23: extern SLPOLYGONLIST  *field_polygonlist;
24: extern SLPOINTLIST    *field_pointlist;
25:
26: extern SLPOLYGONLIST  *collide_polygonlist;
27: extern SLPOINTLIST    *collide_pointlist;
28:
29: void  create_field();   /* testcourse.c */
30: void  destroy_field();  /* testcourse.c */
31:
32: SLPOLYGONLIST *car_polygonlist;
33: SLPOINTLIST   *car_pointlist;
34: SLPOLYGONLIST *fwheel_polygonlist;
35: SLPOINTLIST   *fwheel_pointlist;
36:
37: SLTRANSWORK   *work;
38: SLMINMAX      *minmaxfull, *minmaxt;
39: SLPARAMETER   parameter;
40:
41: typedef struct {
42:   int width;           /* ボディ全幅 */
43:   int length;          /* ボディ全長 */
44:   int height;          /* ボディ全高 */
45:   int fshaft;          /* Fホイール幅 */
46:   int rshaft;          /* Rホイール幅 */
47:   int fwidth;          /* Fタイヤ幅 */
48:   int fradius;         /* Fタイヤ半径 */
49:   int rwidth;          /* Rタイヤ幅 */
50:   int rradius;         /* Rタイヤ半径 */
51:   int wheelbase;       /* ホイールベース */
52:   int steeringratio;   /* ステアリング係数 */
53:   int accel;           /* 加速度 */
54:   int brake;           /* 減速度 */
55: } CarInfo;
56:
57: /* 速度の補正値(大きくするほど加減速が極端になる) */
58: #define RUNFIX  0.2
59:
60: /* 車 */
61: void  create_car( c )
62: CarInfo *c;
63: {
64:   car_polygonlist = malloc( sizeof(SLPOLYGONLIST)+sizeof(SLPOLYGON
)*18 );
65:   car_pointlist = malloc( sizeof(SLPOINTLIST)+sizeof(SLPOINT)*24 )
;
66:   car_polygonlist->n = 0;
67:   car_pointlist->n = 0;
68:   /* 車体 */
69:   makebox( car_polygonlist, car_pointlist,
70:       -c->width/2, -c->height, -c->length/2,
71:        c->width/2, 0,           c->length/2,
72:       &std_red );
73:   /* 後輪(車体に固定されているため同一のポリゴンリストに入れる) */
74:   makebox( car_polygonlist, car_pointlist,
75:       -c->fshaft/2-c->rwidth/2, -c->height, -c->wheelbase/2-c->rra
dius,
```

▶たぶんうちの生徒が世界一Oh!Xを買っていると思う。　　青木　俊充(20)東京都

```
  76:         -c->fshaft/2+c->rwidth/2, 0,              -c->wheelbase/2+c->rra
dius,
  77:         &std_darkgray );
  78:   makebox( car_polygonlist, car_pointlist,
  79:         c->fshaft/2-c->rwidth/2, -c->rradius*2, -c->wheelbase/2-c->r
radius,
  80:         c->fshaft/2+c->rwidth/2, 0,               -c->wheelbase/2+c->r
radius,
  81:         &std_darkgray );
  82:   AddNorm( car_polygonlist, car_pointlist );
  83:   SortPoly( car_polygonlist, car_pointlist );
  84:   /* 前輪(車体から独立しているため別のポリゴンリストに入れる) */
  85:   fwheel_polygonlist = malloc( sizeof(SLPOLYGONLIST)+sizeof(SLPOLY
GON)*6 );
  86:   fwheel_pointlist = malloc( sizeof(SLPOINTLIST)+sizeof(SLPOINT)*8
 );
  87:   fwheel_polygonlist->n = 0;
  88:   fwheel_pointlist->n = 0;
  89:   makebox( fwheel_polygonlist, fwheel_pointlist,
  90:        -c->rwidth/2, -c->fradius*2, -c->fradius,
  91:         c->rwidth/2, 0,              c->fradius,
  92:         &std_darkgray );
  93:   AddNorm( fwheel_polygonlist, fwheel_pointlist );
  94:   return;
  95: }
  96:
  97: void  destroy_car()
  98: {
  99:   free( car_polygonlist );
 100:   free( car_pointlist );
 101:   free( fwheel_polygonlist );
 102:   free( fwheel_pointlist );
 103:   return;
 104: }
 105:
 106: VECTOR3 v, va, vb, vc;                       /* 視点の位置と基底ベクトル */
 107: VECTOR3 b[2], ba[2], bb[2], bc[2];   /* 車体の位置と基底ベクトル */
 108: VECTOR3 w, wa, wb, wc;                       /* 前輪の位置と基底ベクトル */
 109: VECTOR3 l;                                   /* 光源の方向ベクトル */
 110:
 111: double  radius, centerx, centery, centerz, run, co, si;
 112: int     sp, time = 0;
 113: int     mscur, x, y, msdt, lb, rb;
 114: double  theta = 0.0;        /* ステアリング角 */
 115: int     t1, t2, dt;         /* 時刻と時差 */
 116: int     velocity = 0;       /* 速度 */
 117: int     viewmode = 1;       /* 視点モード */
 118: int     p = 0;
 119:
 120: /* 車の情報 */
 121: static CarInfo  testcar = {
 122:   20, /* ボディ全幅 */
 123:   50, /* ボディ全長 */
 124:   12, /* ボディ全高 */
 125:   30, /* Fホイール幅 */
 126:   32, /* Rホイール幅 */
 127:   5,  /* Fタイヤ幅 */
 128:   5,  /* Fタイヤ半径 */
 129:   6,  /* Rタイヤ幅 */
 130:   6,  /* Rタイヤ半径 */
 131:   40, /* ホイールベース */
 132:   2,  /* ステアリング係数 */
 133:   1,  /* 加速度 */
 134:   2 /* 減速度 */
 135: };
 136:
 137: SLPOLYGONLIST *dpypolygonlist;
 138: SLPOINTLIST   *dpypointlist;
 139: int           dpymode = 0;
 140: CHECKINFOLIST *collide_checkinfolist;
 141:
 142: void  drive()
 143: {
 144:   /* 前回の時刻との差 */
 145:   t2 = ONTIME();
 146:   dt = TIMEDIFFERENCE(t2,t1);
 147:   t1 = t2;
 148:   /* 速度 */
 149:   /*
 150:   右ボタン  アクセル
 151:   左ボタン  ブレーキ
 152:   両ボタン  減速および後退
 153:   */
 154:   if ( lb && rb ) {
 155:     velocity -= testcar.brake;
 156:   } else if ( rb ) {
 157:     velocity += testcar.accel;
 158:   } else if ( lb ) {
 159:     if ( velocity > 0 ) {
 160:       velocity -= testcar.brake;
 161:       if ( velocity < 0 ) velocity = 0;
 162:     } else if ( velocity < 0 ) {
 163:       velocity += testcar.brake;
 164:       if ( velocity > 0 ) velocity = 0;
 165:     }
 166:   }
 167:   /* 速度に前回の時刻との差をかければ移動量が出る */
 168:   run = (double)(velocity*dt)*RUNFIX;
 169:
 170:   theta = ITOD((128-x)*testcar.steeringratio);  /* ステアリング角 */
 171:   if ( velocity == 0 ) {
 172:      b[1-p][0] =  b[p][0];  b[1-p][1] =  b[p][1];  b[1-p][2] =  b[
p][2];
 173:     ba[1-p][0] = ba[p][0]; ba[1-p][1] = ba[p][1]; ba[1-p][2] = ba[
p][2];
 174:     bb[1-p][0] = bb[p][0]; bb[1-p][1] = bb[p][1]; bb[1-p][2] = bb[
p][2];
 175:     bc[1-p][0] = bc[p][0]; bc[1-p][1] = bc[p][1]; bc[1-p][2] = bc[
p][2];
 176:   } else {
 177:     /* 基底座標系のγα平面上の極低速円旋回 */
 178:     if ( x < 128 ) {  /* 左折 */
 179:       radius = testcar.wheelbase / tan( theta );          /* 旋回半径
*/
 180:       centerx = b[p][0] - bc[p][0]*testcar.wheelbase/2  /* 旋回中心
*/
 181:             - ba[p][0]*radius;
 182:       centery = b[p][1] - bc[p][1]*testcar.wheelbase/2
 183:             - ba[p][1]*radius;
 184:       centerz = b[p][2] - bc[p][2]*testcar.wheelbase/2
 185:             - ba[p][2]*radius;
 186:       /* 旋回中心を中心として車体を回転する */
 187:       co = radius / sqrt( radius*radius + run*run );
 188:       si = run / sqrt( radius*radius + run*run );
 189:       /* 位置ベクトルは旋回中心からの相対位置ベクトルを回転する */
 190:       b[1-p][0] = centerx + radius*co*ba[p][0] + radius*si*bc[p][0
];
 191:       b[1-p][1] = centery + radius*co*ba[p][1] + radius*si*bc[p][1
];
 192:       b[1-p][2] = centerz + radius*co*ba[p][2] + radius*si*bc[p][2
];
 193:       /* 基底ベクトルはそのものを回転する */
 194:       /* bb(β軸)は旋回運動では変化しない */
 195:       ba[1-p][0] =  co*ba[p][0] + si*bc[p][0];
 196:       bc[1-p][0] = -si*ba[p][0] + co*bc[p][0];
 197:
 198:       ba[1-p][1] =  co*ba[p][1] + si*bc[p][1];
 199:       bc[1-p][1] = -si*ba[p][1] + co*bc[p][1];
 200:
 201:       ba[1-p][2] =  co*ba[p][2] + si*bc[p][2];
 202:       bc[1-p][2] = -si*ba[p][2] + co*bc[p][2];
 203:
 204:       b[1-p][0] += bc[1-p][0]*testcar.wheelbase/2;
 205:       b[1-p][1] += bc[1-p][1]*testcar.wheelbase/2;
 206:       b[1-p][2] += bc[1-p][2]*testcar.wheelbase/2;
 207:     } else if ( x > 128 ) { /* 右折 */
 208:       radius = testcar.wheelbase / tan( -theta );
 209:       centerx = b[p][0] - bc[p][0]*testcar.wheelbase/2
 210:             + ba[p][0] * radius;
 211:       centery = b[p][1] - bc[p][1]*testcar.wheelbase/2
 212:             + ba[p][1] * radius;
 213:       centerz = b[p][2] - bc[p][2]*testcar.wheelbase/2
 214:             + ba[p][2] * radius;
 215:       co = radius / sqrt( radius*radius + run*run );
 216:       si = run / sqrt( radius*radius + run*run );
 217:       b[1-p][0] = centerx - radius*co*ba[p][0] + radius*si*bc[p][0
];
 218:       b[1-p][1] = centery - radius*co*ba[p][1] + radius*si*bc[p][1
];
 219:       b[1-p][2] = centerz - radius*co*ba[p][2] + radius*si*bc[p][2
];
 220:
 221:       ba[1-p][0] = co*ba[p][0] - si*bc[p][0];
 222:       bc[1-p][0] = si*ba[p][0] + co*bc[p][0];
 223:
 224:       ba[1-p][1] = co*ba[p][1] - si*bc[p][1];
 225:       bc[1-p][1] = si*ba[p][1] + co*bc[p][1];
 226:
 227:       ba[1-p][2] = co*ba[p][2] - si*bc[p][2];
 228:       bc[1-p][2] = si*ba[p][2] + co*bc[p][2];
 229:
 230:       b[1-p][0] += bc[1-p][0]*testcar.wheelbase/2;
 231:       b[1-p][1] += bc[1-p][1]*testcar.wheelbase/2;
 232:       b[1-p][2] += bc[1-p][2]*testcar.wheelbase/2;
 233:     } else {  /* 直進 */
 234:       b[1-p][0] = b[p][0] + bc[p][0] * run;
 235:       b[1-p][1] = b[p][1] + bc[p][1] * run;
 236:       b[1-p][2] = b[p][2] + bc[p][2] * run;
 237:       ba[1-p][0] = ba[p][0]; ba[1-p][1] = ba[p][1]; ba[1-p][2] = b
a[p][2];
 238:       bb[1-p][0] = bb[p][0]; bb[1-p][1] = bb[p][1]; bb[1-p][2] = b
b[p][2];
 239:       bc[1-p][0] = bc[p][0]; bc[1-p][1] = bc[p][1]; bc[1-p][2] = b
c[p][2];
 240:     }
 241:   }
 242:   return;
 243: }
 244:
 245: void  checkcollision()
 246: {
 247:   int     i, ry, rd;
 248:   double  x, y, z, l;
 249:
 250:   /* 検索起点 */
 251:   x = b[1-p][0] - (testcar.height*0.5)*bb[1-p][0];
 252:   y = b[1-p][1] - (testcar.height*0.5)*bb[1-p][1];
 253:   z = b[1-p][2] - (testcar.height*0.5)*bb[1-p][2];
 254:   i = checks2( &ry, &rd, collide_checkinfolist, (int)x, (int)y, (i
nt)z );
 255:   if ( i == -1 ) return;
 256:   /* 壁にぶつかったら前の状態に戻す */
 257:   if ( collide_polygonlist->polygon[i].palet == &std_magenta ) {
 258:      b[1-p][0] =  b[p][0];  b[1-p][1] =  b[p][1];  b[1-p][2] =  b[
p][2];
 259:     ba[1-p][0] = ba[p][0]; ba[1-p][1] = ba[p][1]; ba[1-p][2] = ba[
p][2];
 260:     bb[1-p][0] = bb[p][0]; bb[1-p][1] = bb[p][1]; bb[1-p][2] = bb[
p][2];
 261:     bc[1-p][0] = bc[p][0]; bc[1-p][1] = bc[p][1]; bc[1-p][2] = bc[
p][2];
 262:     return;
 263:   }
 264:   /* 新しい位置(y座標のみ変わる) */
 265:   b[1-p][1] = (double)ry;
 266:   /* 面法線を求める(checklib本体に組み込むのがスジか?) */
 267:   x = -(double)collide_checkinfolist->ci[i].a;
 268:   y = -(double)collide_checkinfolist->ci[i].b;
 269:   z = -(double)collide_checkinfolist->ci[i].c;
 270:   l = sqrt( x*x + y*y + z*z );
 271:   x /= l;
 272:   y /= l;
```


▶ゲームの怪音(?)シリーズ。「餓狼伝説スペシャル」のギースの「疾風拳」は「しっとるけぇ」と聴こえる。　西村　正明(22)東京都

```
273:     z /= 1;
274:     /* 新しいβ軸(面法線) */
275:     bb[1-p][0] = x;
276:     bb[1-p][1] = y;
277:     bb[1-p][2] = z;
278:     /* 新しいγ軸(補正前のγ軸から新β軸成分を取り除いて正規化したもの) */
279:     l = x*bc[1-p][0] + y*bc[1-p][1] + z*bc[1-p][2];
280:     x = bc[1-p][0] - l*x;
281:     y = bc[1-p][1] - l*y;
282:     z = bc[1-p][2] - l*z;
283:     l = sqrt( x*x + y*y + z*z );
284:     bc[1-p][0] = x/l;
285:     bc[1-p][1] = y/l;
286:     bc[1-p][2] = z/l;
287:     /* 新しいα軸(新β軸と新γ軸の外積) */
288:     ba[1-p][0] = bb[1-p][1]*bc[1-p][2] - bb[1-p][2]*bc[1-p][1];
289:     ba[1-p][1] = bb[1-p][2]*bc[1-p][0] - bb[1-p][0]*bc[1-p][2];
290:     ba[1-p][2] = bb[1-p][0]*bc[1-p][1] - bb[1-p][1]*bc[1-p][0];
291:     return;
292: }
293:
294: void  display()
295: {
296:     if ( time%2 == 0 ) {
297:       SetWritePlane( (unsigned short *)0xC00200 );
298:       SetClearPlane( (unsigned short *)0xC00200 );
299:     } else {
300:       SetWritePlane( (unsigned short *)0xC00000 );
301:       SetClearPlane( (unsigned short *)0xC00000 );
302:     }
303:     ClearBox( minmaxfull );
304:
305:     /* 視点/視線 */
306:     if ( viewmode == 0 ) {
307:       v[0] =  0.0; v[1]  = -1500.0;        v[2]  = -2000.0;         /*
位置 */
308:       va[0] = 1.0; va[1] = 0.0;            va[2] = 0.0;             /*
α軸 */
309:       vb[0] = 0.0; vb[1] = sqrt(2.0)/2.0; vb[2] = -sqrt(2.0)/2.0; /*
β軸 */
310:       vc[0] = 0.0; vc[1] = sqrt(2.0)/2.0; vc[2] = sqrt(2.0)/2.0;  /*
γ軸 */
311:     } else {
312:       /* 位置 */
313:       v[0] = b[1-p][0] - (testcar.height*1.5)*bb[1-p][0];
314:       v[1] = b[1-p][1] - (testcar.height*1.5)*bb[1-p][1];
315:       v[2] = b[1-p][2] - (testcar.height*1.5)*bb[1-p][2];
316:       va[0] = ba[1-p][0]; va[1] = ba[1-p][1]; va[2] = ba[1-p][2]; /*
α軸 */
317:       vb[0] = bb[1-p][0]; vb[1] = bb[1-p][1]; vb[2] = bb[1-p][2]; /*
β軸 */
318:       vc[0] = bc[1-p][0]; vc[1] = bc[1-p][1]; vc[2] = bc[1-p][2]; /*
γ軸 */
319:     }
320:     /* 光源の方向ベクトル */
321:     l[0] = -1.0; l[1] = -3.0; l[2] = -2.0;
322:     /* 地面の表示 */
323:     eulerA2G( &parameter, &v, &va, &vb, &vc, &l );
324:     TranslateAll( &parameter, work, dpypointlist, minmaxt );
325:     DisplayPolygonList( dpypolygonlist, work, minmaxt );
326:
327:     /* 車体の表示 */
328:     eulerA2A( &parameter, &v, &va, &vb, &vc,
329:               &b[1-p], &ba[1-p], &bb[1-p], &bc[1-p] );
330:     TranslateAll( &parameter, work, car_pointlist, minmaxt );
331:     DisplayPolygonList( car_polygonlist, work, minmaxt );
332:
333:     /* 前輪 */
334:     /* α軸 */
335:     wa[0] = ba[1-p][0]*cos(theta) + bc[1-p][0]*sin(theta);
336:     wa[1] = ba[1-p][1]*cos(theta) + bc[1-p][1]*sin(theta);
337:     wa[2] = ba[1-p][2]*cos(theta) + bc[1-p][2]*sin(theta);
338:     /* β軸 */
339:     wb[0] = bb[1-p][0];
340:     wb[1] = bb[1-p][1];
341:     wb[2] = bb[1-p][2];
342:     /* γ軸 */
343:     wc[0] = -ba[1-p][0]*sin(theta) + bc[1-p][0]*cos(theta);
344:     wc[1] = -ba[1-p][1]*sin(theta) + bc[1-p][1]*cos(theta);
345:     wc[2] = -ba[1-p][2]*sin(theta) + bc[1-p][2]*cos(theta);
346:     /* 位置 */
347:     w[0] =  b[1-p][0] + ba[1-p][0]*testcar.fshaft/2 +
348:                         bc[1-p][0]*testcar.wheelbase/2;
349:     w[1] =  b[1-p][1] + ba[1-p][1]*testcar.fshaft/2 +
350:                         bc[1-p][1]*testcar.wheelbase/2;
351:     w[2] =  b[1-p][2] + ba[1-p][2]*testcar.fshaft/2 +
352:                         bc[1-p][2]*testcar.wheelbase/2;
353:     /* 右前輪の表示 */
354:     eulerA2A( &parameter, &v, &va, &vb, &vc, &w, &wa, &wb, &wc ),
355:     TranslateAll( &parameter, work, fwheel_pointlist, minmaxt );
356:     DisplayPolygonList( fwheel_polygonlist, work, minmaxt );
357:
358:     /* 左前輪 */
359:     /* 位置 */
360:     w[0] =  b[1-p][0] - ba[1-p][0]*testcar.fshaft/2 +
361:                         bc[1-p][0]*testcar.wheelbase/2;
362:     w[1] =  b[1-p][1] - ba[1-p][1]*testcar.fshaft/2 +
363:                         bc[1-p][1]*testcar.wheelbase/2;
364:     w[2] =  b[1-p][2] - ba[1-p][2]*testcar.fshaft/2 +
365:                         bc[1-p][2]*testcar.wheelbase/2;
366:     /* 左前輪の表示 */
367:     eulerA2A( &parameter, &v, &va, &vb, &vc, &w, &wa, &wb, &wc );
368:     TranslateAll( &parameter, work, fwheel_pointlist, minmaxt );
369:     DisplayPolygonList( fwheel_polygonlist, work, minmaxt );
370:     return;
371: }
372:
373: int main()
374: {
375:     create_field();
376:     collide_checkinfolist =
377:         createCheckInfoList( collide_polygonlist->n,
378:                              collide_polygonlist,
379:                              collide_pointlist );
380:     generateCheckInfoList( collide_checkinfolist );
381:
382:     create_car( &testcar );
383:
384:     work = malloc( sizeof(SLTRANSWORK)*N_POINT );
385:     minmaxfull = malloc( sizeof(SLMINMAX)*2 );
386:     minmaxt = malloc( sizeof(SLMINMAX)*2 );
387:
388:     /* いつでも全画面クリアする */
389:     minmaxfull[0].xmin = 0;
390:     minmaxfull[0].ymin = 0;
391:     minmaxfull[0].xmax = 255;
392:     minmaxfull[0].ymax = 255;
393:     minmaxfull[1].xmin = -1;
394:     minmaxfull[1].ymin = -1;
395:     minmaxfull[1].xmax = -1;
396:     minmaxfull[1].ymax = -1;
397:
398:     CRTMOD( 14 );
399:     G_CLR_ON();
400:     B_CUROFF();
401:     MS_INIT();
402:     MS_CUROF();
403:     MS_LIMIT(0,0,255,255);
404:     MS_CURST(128,128);
405:     MS_CUROF();
406:     SKEY_MOD(0,0,0);
407:     sp = SUPER( 0 );
408:
409:     SetClearColor( 0x00100010 );
410:     SetWindowSize( 256, 256 );
411:     SetWindowCenter( 128, 128 );
412:
413:     /* 車体 */
414:     p = 0;
415:     /* 位置(ワールド座標) */
416:     b[p][0]  = 0.0; b[p][1]  = 0.0; b[p][2]  = 0.0;
417:     /* 基底ベクトル(α軸) */
418:     ba[p][0] = 1.0; ba[p][1] = 0.0; ba[p][2] = 0.0;
419:     /* 基底ベクトル(β軸) */
420:     bb[p][0] = 0.0; bb[p][1] = 1.0; bb[p][2] = 0.0;
421:     /* 基底ベクトル(γ軸) */
422:     bc[p][0] = 0.0; bc[p][1] = 0.0; bc[p][2] = 1.0;
423:
424:     /* 経過時間を計測する */
425:     t1 = ONTIME();
426:
427:     /* 表示用のフィールド */
428:     dpypolygonlist = field_polygonlist;
429:     dpypointlist = field_pointlist;
430:
431:     /* メインループ */
132:     for (;;) {
433:       /* マウス入力 */
434:       mscur = MS_CURGT();
435:       x = mscur/65536;
436:       y = mscur%65536;
437:       msdt = MS_GETDT()%65536;
438:       lb = msdt/256;
439:       rb = msdt%256;
440:       /* ESCキーで終了 */
441:       if ( BITSNS(0x00)&2 ) break;
442:       /* F1で視点モード切り替え */
443:       if ( BITSNS(0x0C)&8 ) {
444:         while ( BITSNS(0x0C)&8 );
445:         viewmode = 1 - viewmode;
446:       }
447:       if ( BITSNS(0x0C)&16 ) {
448:         while ( BITSNS(0x0C)&16 );
449:         dpymode = 1 - dpymode;
450:         if ( dpymode == 0 ) {
451:           dpypolygonlist = field_polygonlist;
452:           dpypointlist = field_pointlist;
453:         } else {
454:           dpypolygonlist = collide_polygonlist;
455:           dpypointlist = collide_pointlist;
456:         }
457:       }
458:
459:       drive();
460:       checkcollision();
461:       display();
462:
463:       if ( time%2 == 0 ) HOME( 0, 256, 0 );
464:         else     HOME( 0, 0, 0 );
465:       time++;
466:
467:       /* phaseを反転する */
468:       p = 1 - p;
469:     }
470:     SUPER( sp );
471:
472:     free( work );
473:     free( minmaxt );
474:     free( minmaxfull );
475:     destroy_car();
476:     destroy_field();
477:
478:     B_CURON();
479:     CRTMOD( 16 );
480:     KFLUSHIO( 0xFF );
481:
482:     return 0;
483: }
```

▶いきなりだが「ジャイアントポッキー」は嫌いだ。1,000円もするくせにほんのりまずい。それだけだけど。
幸　俊威(19)大阪府

SIDE B

とりあえず座標系をマスター

Yokouchi Takeshi 横内 威至

SLASHver.2.0のリリースを控え，目標に向かって邁進している横内氏
今回のキーワードは，行列，オイラー角，そして座標変換
座標系をマスターし，空間をこの手につかもう

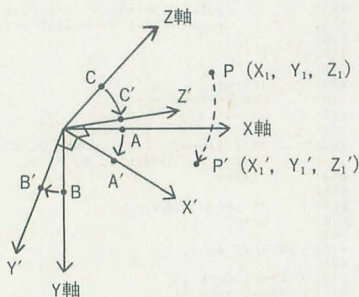
ようやく，「リッジレーサー」をやった。熱すぎる。ドリフトが熱すぎる。ドリフト状態でのグリップ，というかコントロールの容易さは異常だとは思うが，挙動はかなりシビレてる。コーナーに直角に4輪ドリフトなんて狂いすぎてるけど気持ちよすぎる。さらに加速して，逆走状態ドリフトまでしてコーナーをクリアするともう無敵。ギャラリーもかなりウケてくれる。フェイントモーションもやたらとクる。しかしフェイントがらみのコーナーがないのは残念である。できることならFFのテクニックも磨きたい。次はサイドブレーキもつけて，そして車も選べるとうれしい。

おっと，こんな感想なんかはどうでもいい。せめてカウンターで車をネジ伏せる快感を再現できるレベルのものぐらいは作らねばならない。でも俺はまだそんなレベルまでは到達していないからさらにがんばらねばならない。

図1　座標変換

$$R=\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \qquad P=\begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \end{pmatrix}$$

X′軸　Y′軸　Z′軸

$$P'\begin{pmatrix} X'_1 \\ Y'_1 \\ Z'_1 \end{pmatrix}=RP$$

A (1, 0, 0)	A′ (a, d, g)
B (0, 1, 0)	B′ (b, e, h)
C (0, 0, 1)	C′ (c, f, i)

この$\overrightarrow{A'}$，$\overrightarrow{B'}$，$\overrightarrow{C'}$をX′，Y′，Z′軸として座標系を行列Rが示す。

うるさくも座標系を

もっぱらネック扱いしてきた座標系であるが，そろそろみんなの理解が深まってきたようだ。そうなると，座標系よりも，表示関係に新たな問題を抱え込んできている。しかし，SLASH ver.2.0のリリースが遅れてしまっているため問題の提起はあと回し。簡単にいうと，テキストとグラフィックの融合表示がかなりの核心部分となっているのである。

それでは再び座標系周りであるが，本当ならばすでにSLASH ver.2.0とともにそのサンプルが配布されているはずであった。このサンプルが今回説明したいことのよき実例となるはずだったのだが……困った。しかし，ないものはしようがないため，収録予定のサンプルリストを参照して雰囲気をつかんでいただくことにしよう。

さて，座標系を理解するうえでしっかり覚えておいてもらいたいのは，行列のもつ意味だ。以前，ちょっと書いたところに間違いがあった。非常に申しわけない。ここで図1を見てもらいたい。図1の行列Rはある座標系を意味する。ある座標系とは，図1のX′Y′Z′軸のことである。そして座標変換とは，基準となる座標系からこの行列の示す座標系に「乗せ替える」ような感じだ。

しかし，この行列Rによる空間（X′Y′Z′系）は，元のXYZ系の座標で表されている。となれば，むしろ「X′Y′Z′系をXYZ系の空間に投影することにより，X′Y′Z′系での点 (X_1, Y_1, Z_1) からXYZ系の点P′

(X1′,Y1′,Z1′)を得る」と表現するほうが正しいであろう。はっきりとつかむには，X′Y′Z′系を物体座標系，XYZ系をワールド座標系として考えていけばいい。

とりあえず，この行列がワールド座標系に投影された物体座標系を意味し，その第1列がX軸，第2列がY軸，第3列がZ軸を示すことを理解しておいてもらいたい。

いまは混乱を招かないためにも，基準をワールド座標系としておく。このワールド座標系は，このままでは画面座標系と等しい状態である。このワールド座標系を任意の位置から見る，つまり画面座標系と一致しない状況にするにはまだいろいろと考えなければならない。

ここでは，変換行列がワールド座標系に投影された物体の座標系を示すことだけをはっきり認識しておいてもらいたい。

行列に細工する

そして，座標変換の行列にいろいろと細工をすれば，座標変換をもっと拡張できる。行列を構成する3ベクトルが物体の姿勢を決めるものであり，これをうまく制御することが物体の制御であるのだ。

しかし，SLASH ver.2.0でないと行列の直接指定ができないため，システムが手に入るまでしばらく辛抱してほしい。

ちなみにSLASH ver.2.0での行列の扱いであるが，簡単に図2に解説をしておく。以前に丹氏が解説していた，「行列からオイラー角を求められればすべて解決するはず」というオイラー角と行列のやりとりもSLASH ver.2.0ではカバーしている。ただし，行列の定めとして「必ずしも逆行列が存在するとは限らない」のであり，それなりにうまく計算してオイラー角に直さねばならない。

では，このオイラー角と行列の変換ルーチンではどの程度の精度となっているのであろうか。まずはこのルーチンを解説しておこう(図3)。このアルゴリズムで行列からオイラー角が求められるのだが，cosXの値によってヤバイことが起こることを見抜けるだろうか。理論的にはバッチリだが，現在扱っている行列が浮動小数点などという精度の細かいものでないため，このcosXが0に近づくとまともな計算が不可能になってくるのだ。では，そのような状態とはどんな様子だろうか。答えはcosXが0に近い状態，つまり物体が直立した状態である。かなり特殊な状態であるため，それほど害が出ないとは思うが，完璧な動作でない以上はおまけ的なルーチンと考えたほうが無難。どうせ，これを使うのはシェーディングのためだけであろうと願っている。

図2　SLASH ver.2.0での行列の扱い

追加コール「TRANSLATER_MATRIX」について

これは行列を直接指定して変換するルーチン。行列はマトリックスワーク

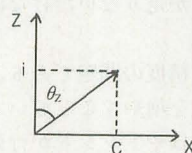
+0	1行1列
+2	1行2列
+4	1行3列
+6	2行1列
⋮	⋮
+16	3行3列
+18	シェーディングデータ1
+20	シェーディングデータ2
+24	シェーディングデータ3
+28 〜 +31	リザーブエリア

を指定することになる。

図3　オイラー角と行列の変換

変換行列$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$は，オイラー角のパラメータ（ZXY順の回転とする）で表すと，

$$R=\begin{pmatrix} \cos Y\cos Z+\sin Y\sin Z & \sin X\sin Y\cos Z-\cos Y\sin Z & \cos X\sin Y \\ \cos X\sin Z & \cos X\cos Z & -\sin X \\ \sin X\cos Y\sin Z-\sin Y\cos Z & \sin X\cos Y\cos Z+\sin Y\sin Z & \cos X\cos Y \end{pmatrix}$$として表される。

求めるべき値をX′，Y′，Z′（それぞれXYZが成す行列に近くなるよう努力する）とすると，

まずZ軸に注目すると，fの値に関係なくY軸方向を見ると左のようになる。θ_ZはYに等しい。

よって，$\sin Y'=i/\sqrt{i^2+c^2}$

$\cos Y'=c/\sqrt{i^2+c^2}$

ただし，$i^2+c^2=0$のときは，$\sin Y=0$，$\cos Y=1$とする。

次に，$\sin X'=-f$

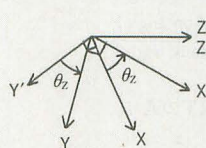
$\cos X'=\sqrt{1-f^2}$

ただし，常に$0\leqq\cos X'$とする。

ここで，X′，Y′については求めた値，Z′=0のときの行列を考えると，

$$R'=\begin{pmatrix} \cos Y' & \sin X'\sin Y' & \cos X'\sin Y' \\ 0 & \cos X' & -\sin X' \\ -\sin Y' & \sin X'\cos Y' & \cos X'\cos Y' \end{pmatrix}$$

となる。

ここでR′とRの関係を考えると，

常にこのようになっている。つまり，バンク角のみの回転で一致する状態となっている。

この時，X軸は（a，d，g），X′軸は（cosY′，0，−sinY′）である。また，$|\vec{X}|$，$|\vec{X'}|$ともに1である。

よって，内積$a\times\cos Y'+d\times 0+g\times(-\sin Y')=\cos Z'$

問題はsinZである。符号が判定しにくいが，$|\sin Z'|=\sqrt{1-\cos^2 Z'}$となる。

符号判定はこのX軸とX′軸の外積がZ軸の方向と合っているかどうかで調べられる。

しかし，ここでR′とRの関係を考え直す。Z′値次第で同じ変換が可能ならば，

X=X′，Y=Y′と考えてかまわない。となると，

$h\times\sin Y'=\sin Y\ (\sin X\cos Y\cos Z+\sin Y\sin Z)=\sin^2 Y\sin Z+\sin X\sin Y\cos Y\cos Z$ ──①

$b\times\cos Y'=\cos Y\ (\sin X\sin Y\cos Z-\cos Y\sin Z)=\cos^2 Y\sin Z+\sin X\sin Y\cos Y\cos Z$ ──②

①−②$=(\sin^2 Y+\cos^2 Y)\ \sin Z$

$=\sin Z$

以上のように手抜きが可能。

このアルゴリズムで苦しいのは$\sqrt{i^2+c^2}\ll 1$の状態である。

遠回りになったが，次に姿勢決定を考える。オイラー角でしか考えないうちは，任意の姿勢からの回転制御は極めて複雑であった。実際，航空力学の本なんかでもこのあたりのことは結構厳しい解説しか行われていない。さっそくその方法を解説しよう(図4)。

まずは航空力学のオイラー角の定義をしっかりととらえてもらおう。一見，SLASHのそれと違うように見える。これは以前説明したように，定義が違っているのである。

しかし，その動作結果はまったく同じであることに気づくであろうか。注意深く考えればSLASHと同じことをやっているのがわかるはず。座標軸の取り方，つまりZ軸を正面とした物体のとらえ方，そしてそのうえでの制御のパターンは完全に一致している。SLASHがあえてこのような座標系を採用していることに納得してもらえたであろう。

では，このオイラー角での自由回転，つまり物体の座標軸に対する回転角とオイラー角について調べることにしよう。これが図4である。なぜこのような式になるかは，実は把握できていない。ある本によれば「明らかに」そうなるらしいが，この複雑な座標軸の絡みからして，どの角度から見ても明らかには思えなかった。私の理解力が足りないだけなのだろうか。

そして確実な問題点。まずは精度の問題である。これに関してはサインテーブルを増やすことでいくらでも上げることができる。もはやトレンドが行列に移行したため，これについてはあまり深くは追求していない。というのも，これも先ほどの行列→オイラー角変換と同じような欠点があり，そしてこちらのほうがかなり悪質であるからだ。というのはsecXの存在である。つまり1/cosXという，範囲を定めることのできない値が必要となっているからなのである。

これについてはもうどうしようもないと思う。やはり物体が垂直に近いときにその問題が表れるが，動作はかなり致命的。よほどの解決策を思いつかないかぎりは，オイラー角による制御は切り捨てる方向でいきたい。オイラー角は扱いが楽で，このような方法で回転を制御するには計算も少ないため結構旨味はあるのだが，以上の問題によりSLASHには応用したくない。

それでは行列による自由回転制御を考える。変換行列が物体座標系の3軸を意味すること，その3軸は3つのベクトルであったことを思い出してもらいたい。そして，ベクトルは座標によって表される。

また，座標変換と同じように扱って，これらを行列によって変換することが，結局は物体の回転であることに気づいてもらいたい。図1のとおり，各列がX，Y，Z軸を示す座標である。(0,0,0) から，その座標がある方向に座標軸を取るということになる。よって，この座標を狙って回転させれば，任意の姿勢での制御ができるようになるとしつこくいっておく。ということで図5がいよいよそのシステムだ。この姿勢の行列に，左から回転行列を掛ければ「姿勢行列が示す姿勢」をそのままさらにオイラー角の定義に従ってワールド座標系で回転させることになる。逆に右から回転行列を掛ければ，「姿勢行列が示す姿勢」からオイラー角の定義に従って回転させる。

つまりその姿勢からのヘッド，ピッチ，バンクとなるのである。ではそのサンプルリスト（リスト1～3）である。申しわけないが，SLASH ver.2.0のリリースより先になってしまっている。これはそのときについてくるテスタロッサのサンプルの3つの部分，TSTINIT.S，TSTMAIN.S，MOVE.Sを差し換えればOK。まずこのサンプルを起動すると前者の変換，図5でのR'=RRmの変換である。次にBSを押してみよう。「AXIS ROT」と表示されたときは図5のR'=RmRの変換となっている。違いをしっかりと把握してもらいたい。ということなのだが，現段階では動かすこともできないため，リストを参照するだけにとどまることになる。

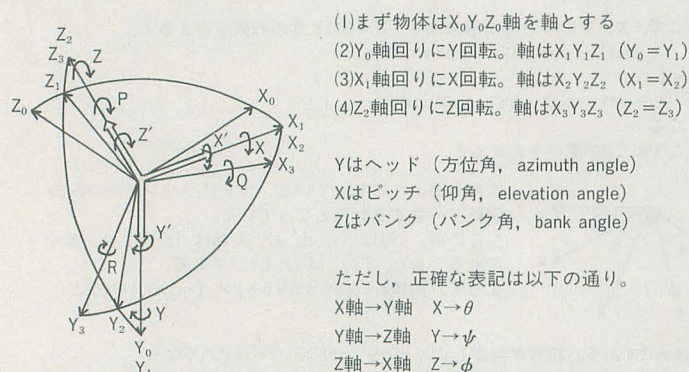
さて，このリストで無駄なことをしている。変換手順は次のようになる。

1)　1回前の行列に回転行列を掛ける
2)　その行列からオイラー角を求める
3)　オイラー角からシェーディングを計算

図4　航空力学のオイラー角モドキ

(1)まず物体は$X_0Y_0Z_0$軸を軸とする
(2)Y_0軸回りにY回転。軸は$X_1Y_1Z_1$（$Y_0=Y_1$）
(3)X_1軸回りにX回転。軸は$X_2Y_2Z_2$（$X_1=X_2$）
(4)Z_2軸回りにZ回転。軸は$X_3Y_3Z_3$（$Z_2=Z_3$）

Yはヘッド（方位角，azimuth angle）
Xはピッチ（仰角，elevation angle）
Zはバンク（バンク角，bank angle）

ただし，正確な表記は以下の通り。
X軸→Y軸　$X\to\theta$
Y軸→Z軸　$Y\to\psi$
Z軸→X軸　$Z\to\phi$

ここで，図の状態でのオイラー角速度X'，Y'，Z'（X_2，Y_1，Z_3軸の角速度）と物体の角速度Q，R，P（X_3，Y_3，Z_3軸の角速度）は，

$$\begin{pmatrix} Z' \\ X' \\ Y' \end{pmatrix} = \begin{pmatrix} 1 & \sin Z\tan X & \cos Z\tan X \\ 0 & \cos Z & -\sin Z \\ 0 & \sin Z\sec X & \cos Z\sec X \end{pmatrix} \begin{pmatrix} P \\ Q \\ R \end{pmatrix}$$

として表される。
正確な表記では，$X'\to\dot{\theta}$，$Y'\to\dot{\psi}$，$Z'\to\dot{\phi}$である。

4)　このオイラー角から行列を作る

以上だが，4)の手順が無駄になることは明白。ということでESCキーを押してもらいたい。テキスト版では手順4)を省いている。

さて，SLASH ver.2.0を手に入れてこのサンプルを実行したとすると，かなりヤバイ状況になるはずだ。手順4)さえ省かなければ正常な動作だったはずなのに。

理由ははっきりしていて，行列からオイラー角を求める段階で計算の誤差が蓄積していくからなのである。ここまで顕著に誤差が表れるのは，なんの色気もなくワード単位で計算してしまったからである。問題が計算精度だとしたら，ワード単位演算を，浮動小数点にしたとすれば平気だろうか。

しかし，やはり誤差が積もっていく。どのようにしても，この誤差を取り除くために適当な補正が必須だ。やらなければ，ルマン24時間なんかをやったとして，ゴールの頃には車がイカレてるなんてことがあり得る。さすがに浮動小数点を導入すればあまり問題ないようではあるが，結構気分の悪い誤差であり，補正についての方法を考えなければならないであろう。

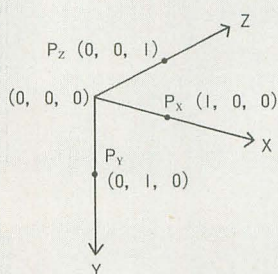
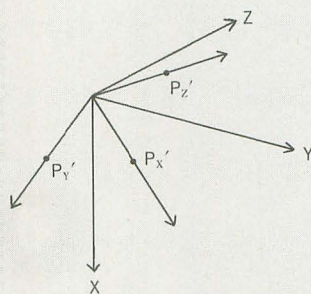
まず最も単純な補正は，先ほどの手順4)である。が，単純ゆえかどうかその問題は大きい。行列→オイラー角の段階での誤差が困るのだ。ステアを切らずに車が回ってしまったり，ストレートで突然ドリフトを始めたりする可能性がある。ということでこれは精密な補正の候補から外しておきたい。となると，まずはっきりすることはＺ軸，物体の方向を決定する軸はかなり重要であることだ。クソ真面目に補正するならば，毎回このＺ軸の長さを１にするように調整する。これは平方根を使うだけで簡単に解決する。となれば問題はその先の軸をどのようにして補正するかだ。１ターン前の行列は正しいと信じれば，とりあえず計算された行列もある程度は正しい。ただし，微妙に３軸の関係が狂っている程度なのだ。だから，ある程度これらを信じて，補正したＺ軸とＹ軸の外積を計算してはどうだろう。それを長さ１に調整すれば，それをＸ軸として使うことができるのではないだろうか。さらに同じようにＹ軸を求めれば，かなり精密な補正となることであろう。

あまりに硬い手口だから処理速度はかなり渋い。もう少し考え直す必要があるが，まだ試してはいないので，とりあえず問題を提起するだけでいまはやめておこうと思う。なぜなら，どの程度の精度が必要で，計算のために裏でもっておくパラメータをロングワードにするか，浮動小数点にしなければならないのかはっきりとは見えていないからである。まあ，とにかくこの問題があることだけをはっきりとさせておこう。

だぶんこれで座標系は見極め

これでいよいよ空間を制御できた感じがする。わかってしまえば結構単純だったような気もするが，なにを基準としていたかがなかなか判別しにくい。特にワールド座標系の変換なんかは混乱しやすい変換になっているので注意しよう。

というあたりで来月はSLASH ver.2.0もリリースできそうなので，マップシステムの解説に入っていきたいと思っている。こいつもかなりシビレるから嫌だ。対象によっても自在に対応しなければならないし，もしかしたらSLASHシステム自体のリメイクを要求しかねない。以前考えていた手前からの描画方法である。そのへんはまた考えよう。最近，緑一色をツモったからもしかしたら死んでるかもしれないので，死んでなければまた来月。それでは，ごきげんよう。

図5　座標変換の原理

このような姿勢とは，$\overrightarrow{P_X}$，$\overrightarrow{P_Y}$，$\overrightarrow{P_Z}$を３軸としたものである。

これも同じく，$\overrightarrow{P_X'}$，$\overrightarrow{P_Y'}$，$\overrightarrow{P_Z'}$を3軸としている。
P_X'，P_Y'，P_Z'は座標を示している。

よって，この座標のなにかを基準にして回転させることで物体を回転させる行列を作ることができる。

$P'=RP$の形は，ワールド座標を基準として回転させる。

つまり，$P'_X=RP_X$，$P_Y'=RP_Y$，$P'_Z=RP_Z$とすれば，$P_XP_YP_Z$で表された物体をワールド座標で回転させることになる。
いま，R'を新しい姿勢行列，Rを変換行列，R_mを$P_XP_YP_Z$で表される行列とすると，

$$R'=RR_m$$

は以上のような変換行列を意味する。

逆に，

$$R'=R_mR$$

は，Rで示される姿勢（物体座標系がワールド座標系と一致しているならば，このRの変換は角速度の変換に等しいと考えられる）をR_mの座標系に乗せることになる。

■リスト1

```
  1:         tst.w   GTR
  2:         beq     ptestgr                  *グラフィック?テキスト?
  3:
  4:         move.w  sRX,d0                   *ウインドウセット
  5:         move.w  sRY,d1
  6:         jsr     T_SETWINDOW
  7:         move.w  sRX,d0
  8:         move.w  sRY,d1
  9:         jsr     TR_SETWINDOW
 10:         move.w  sRX,d0
 11:         move.w  sRY,d1
 12:         asr.w   #1,d0
 13:         asr.w   #1,d1
 14:         jsr     T_SETWINDOWCENTER
 15:         move.w  sRX,d0
 16:         move.w  sRY,d1
 17:         asr.w   #1,d0
 18:         asr.w   #1,d1
 19:         jsr     TR_SETWINDOWCENTER
 20:
 21:         lea.l   PARAMET(pc),a6
 22:         move.w  6(a6),d5
 23:         move.w  8(a6),d6
 24:         move.w  10(a6),d7
 25:         move.w  12(a6),d0
 26:         move.w  14(a6),d1
 27:         jsr     T_GETRAY                 *パラメータからシェーディングデータを
 28:         lea.l   MTWORK(pc),a2
 29:         move.w  d1,18(a2)
 30:         move.w  d0,20(a2)
 31:         move.l  d2,22(a2)
 32:
 33:         lea.l   PARAMET(pc),a6
 34:         lea.l   WORK(pc),a5
 35:         lea.l   POINTLIST(pc),a4
 36:         lea.l   MTWORK(pc),a2
 37:         jsr     T_TRANSLATER_MATRIX      *行列により座標変換
 38:
 39:         lea.l   POLYLISTALL(pc),a6       *ラスタ抜きかどうか
 40:         lea.l   WORK(pc),a5
 41:         tst.w   RASTER
 42:         bne     tptest1
 43:         jsr     T_DRAWPOLYSORT
 44:         bra     tptest11
 45: tptest1:
 46:         jsr     TR_DRAWPOLYSORT
 47: tptest11:
 48:         bra     mainend
 49: ptestgr:
 50:         move.w  sRX,d0              *ウインドウセット
 51:         move.w  sRY,d1
 52:         jsr     SETWINDOW
 53:         move.w  sRX,d0
 54:         move.w  sRY,d1
 55:         jsr     R_SETWINDOW
 56:         move.w  sRX,d0
 57:         move.w  sRY,d1
 58:         asr.w   #1,d0
 59:         asr.w   #1,d1
 60:         jsr     SETWINDOWCENTER
 61:         move.w  sRX,d0
 62:         move.w  sRY,d1
 63:         asr.w   #1,d0
 64:         asr.w   #1,d1
 65:         jsr     R_SETWINDOWCENTER
 66:
 67:         lea.l   MTWORK(pc),a2            *行列からオイラー角三角関数へ
 68:         jsr     MAT_TO_sc
 69:         lea.l   PARAMET(pc),a6
 70:         move.w  d2,a5                    *D2セーブ
 71:         move.w  d2,d0
 72:         move.w  d3,d1
 73:         jsr     GETARC                   *ヘッドのオイラー角へ
 74:         move.w  d0,6(a6)
 75:         move.w  d4,d0
 76:         move.w  d5,d1
 77:         jsr     GETARC                   *ピッチのオイラー角へ
 78:         move.w  d0,8(a6)
 79:         move.w  d6,d0
 80:         move.w  d7,d1
 81:         jsr     GETARC                   *バンクのオイラー角へ
 82:         move.w  d0,10(a6)
 83:         move.w  a5,d2                    *D2ロード
 84:         move.w  12(a6),d0
 85:         move.w  14(a6),d1
 86:         jsr     GETRAYsc                 *シェーディングデータ
 87:         move.w  d1,18(a2)
 88:         move.w  d0,20(a2)
 89:         move.l  d2,22(a2)
 90:
 91:         move.w  6(a6),d5                 *オイラー角から行列を求める
 92:         move.w  8(a6),d6                 *手順4に相当する
 93:         move.w  10(a6),d7
 94:         jsr     GETMATRIX
 95:
 96:         lea.l   PARAMET(pc),a6
 97:         lea.l   WORK(pc),a5
 98:         lea.l   POINTLIST(pc),a4
 99:         movea.l CLEARHEAD,a3
100:         lea.l   MTWORK(pc),a2
101:         jsr     TRANSLATER_MATRIX        *行列により座標変換
102:
103:         lea.l   POLYLISTALL(pc),a6       *ラスタ抜きか?
104:         lea.l   WORK(pc),a5
105:         movea.l CLEARHEAD,a4
106:         tst.w   RASTER
107:         bne     ptest1
108:         jsr     DRAWPOLYSORT
109:         bra     ptest11
110: ptest1:
111:         jsr     R_DRAWPOLYSORT
112: ptest11:
113:         movea.l CLEARHEAD,a0
114:         jsr     ADJUSTMINIMAX            *ミニマックスワーク補正
115: mainend:
```

■リスト2

```
  1:         lea.l   POINTLIST,a0             *データ補正
  2:         move.w  (a0)+,d2
  3:         subq.w  #1,d2
  4: ser:
  5:         neg.w   (a0)
  6:         move.w  2(a0),d1
  7:         move.w  4(a0),2(a0)
  8:         neg.w   d1
  9:         move.w  d1,4(a0)
 10:
 11:         addq.l  #6,a0
 12:         dbra    d2,ser
 13:
 14:         lea.l   POLYLISTa,a0
 15:         bsr     sergo
 16:         lea.l   POLYLISTb,a0
 17:         bsr     sergo
 18:         bra     serend
 19: sergo:
 20:         move.w  (a0)+,d0
 21:         subq.w  #1,d0
 22: ser2:
 23:         move.w  0(a0),d1
 24:         cmp.w   #4,d1
 25:         beq     ser2end
 26:         cmp.w   #5,d1
 27:         beq     ser2end
 28:         move.w  2(a0),d2
 29:         move.w  4(a0),d3
 30:         move.w  6(a0),d4
 31:         move.w  8(a0),d5
 32:         subq.w  #1,d1
 33:         beq     ser2tet
 34:         move.w  d4,4(a0)
 35:         move.w  d3,6(a0)
 36:         bra     ser2end
 37: ser2tet:
 38:         move.w  d5,4(a0)
 39:         move.w  d4,6(a0)
 40:         move.w  d3,8(a0)
 41: ser2end:
 42:         lea.l   32(a0),a0
 43:         dbra    d0,ser2
 44: sernn:
 45:         rts
 46: serend:
 47:         lea.l   PARAMET(pc),a6
 48:         lea.l   MTWORK(pc),a2
 49:         move.w  6(a6),d5
 50:         move.w  8(a6),d6
 51:         move.w  10(a6),d7
 52:         jsr     T_GETMATRIX              *ワークに初期行列を得る
 53:
 54:         lea.l   JOYWAY,a0                *キー入力に対する回転の各オイラー角
 55:         moveq.l #0,d0                    *ジョイスティックの判定ビットみたいなもの
 56: mjw0:
 57:         moveq.l #0,d1
 58:         moveq.l #0,d2
 59:         moveq.l #0,d3
 60:         btst.l  #0,d0                    *ビット0でヘッド右回り
 61:         beq     mjw1
 62:         add.w   #32,d2
 63: mjw1:
 64:         btst.l  #1,d0                    *ビット1でヘッド左回り
 65:         beq     mjw2
 66:         sub.w   #32,d2
 67: mjw2:
 68:         btst.l  #2,d0                    *ビット2でピッチ上
 69:         beq     mjw3
 70:         add.w   #32,d1
 71: mjw3:
 72:         btst.l  #3,d0                    *ビット3でピッチ下
 73:         beq     mjw4
 74:         sub.w   #32,d1
 75: mjw4:
 76:         btst.l  #4,d0                    *ビット4でバンク時計回り
 77:         beq     mjw5
 78:         add.w   #32,d3
 79: mjw5:
 80:         btst.l  #5,d0                    *ビット5でバンク反時計回り
 81:         beq     mjw6
 82:         sub.w   #32,d3
 83: mjw6:
 84:         move.w  d1,(a0)+                 *オイラー角ヘッド
 85:         move.w  d2,(a0)+                 *オイラー角ピッチ
 86:         move.w  d3,(a0)+                 *オイラー角バンク
 87:         addq.w  #1,d0
 88:         cmp.w   #64,d0                   *計6ビット、2^6=64
 89:         bcs     mjw0
 90:
 91:         moveq.l #0,d0                    *それぞれの回転行列を生成
 92:         lea.l   JOYWAY(pc),a6
 93:         lea.l   APS,a2                   *APSから64方向分の回転行列を
 94: getjs:
 95:         move.w  (a6)+,d5
 96:         move.w  (a6)+,d6
 97:         move.w  (a6)+,d7
 98:         move.w  d0,-(sp)
 99:         jsr     T_GETMATRIX              *ワークに行列を得る
100:         move.w  (sp)+,d0
101:         addq.w  #1,d0
102:         lea.l   32(a2),a2                *次
103:         cmp.w   #64,d0
104:         bcs     getjs
105:         bra     initend
106: JOYWAY:
107:         ds.w    3*128
108: initend:
```

▶「SLASH」が入ってない～!　　岡田　弘一(22)東京都

■リスト3

```
  1: PLMIX    MACRO    bit1,bit2
  2:          .local   plmi1
  3:          .local   plmi2
  4:
  5:          moveq.l  #0,d1
  6:          btst.l   bit1,d0
  7:          beq      plmi1
  8:          addq.w   #1,d1
  9: plmi1:
 10:          btst.l   bit2,d0
 11:          beq      plmi2
 12:          addq.w   #2,d1
 13: plmi2:
 14:          .endm
 15:
 16:          bra      mover
 17: APS2:
 18:          ds.w     16
 19: APS:
 20:          ds.w     16
 21:          ds.w     16*64
 22:
 23: mover:
 24:          move.b   $807,d0              *キーボードにより回転する方向のビットを立てる
 25:          PLMIX    #6,#4
 26:          lsl.w    #2,d1
 27:          move.w   d1,d7
 28:
 29:          PLMIX    #3,#5
 30:          add.w    d1,d7
 31:
 32:          move.b   $80e,d0
 33:          PLMIX    #3,#2
 34:          lsl.w    #4,d1
 35:          add.w    d1,d7
 36:
 37:          mulu.w   #32,d7.              *行列は16ワード分
 38:          lea.l    APS,a0
 39:
 40:          lea.l    APS2,a0              *姿勢行列待避
 41:          lea.l    MTWORK,a1
 42:          move.l   (a1)+,(a0)+
 43:          move.l   (a1)+,(a0)+
 44:          move.l   (a1)+,(a0)+
 45:          move.l   (a1)+,(a0)+
 46:          move.w   (a1)+,(a0)+
 47:
 48:          lea.l    APS2,a0
 49:          lea.l    APS,a1
 50:          adda.l   d7,a1
 51:          tst.w    EUL                  *オイラー角フラグで右からか左からかを判定
 52:          beq      notaa
 53:          exg.l    a0,a1
 54: notaa:
 55:          lea.l    MTWORK(pc),a2        *姿勢行列=姿勢行列*回転行列
 56:                                        *あるいは 回転行列*姿勢行列
 57:          move.w   (a0),d0              *行列の合成
 58:          move.w   6(a0),d1
 59:          move.w   12(a0),d2
 60:          muls.w   (a1)+,d0
 61:          muls.w   (a1)+,d1
 62:          muls.w   (a1)+,d2
 63:          subq.l   #6,a1
 64:          add.l    d0,d1
 65:          add.l    d1,d2
 66:          lsl.l    #2,d2
 67:          swap.w   d2
 68:          move.w   d2,(a2)+
 69:
 70:          move.w   2(a0),d0
 71:          move.w   8(a0),d1
 72:          move.w   14(a0),d2
 73:          muls.w   (a1)+,d0
 74:          muls.w   (a1)+,d1
 75:          muls.w   (a1)+,d2
 76:          subq.l   #6,a1
 77:          add.l    d0,d1
 78:          add.l    d1,d2
 79:          lsl.l    #2,d2
 80:          swap.w   d2
 81:          move.w   d2,(a2)+
 82:
 83:          move.w   4(a0),d0
 84:          move.w   10(a0),d1
 85:          move.w   16(a0),d2
 86:          muls.w   (a1)+,d0
 87:          muls.w   (a1)+,d1
 88:          muls.w   (a1)+,d2
 89:          add.l    d0,d1
 90:          add.l    d1,d2
 91:          lsl.l    #2,d2
 92:          swap.w   d2
 93:          move.w   d2,(a2)+
 94:
 95:          move.w   (a0),d0
 96:          move.w   6(a0),d1
 97:          move.w   12(a0),d2
 98:          muls.w   (a1)+,d0
 99:          muls.w   (a1)+,d1
100:          muls.w   (a1)+,d2
101:          subq.l   #6,a1
102:          add.l    d0,d1
103:          add.l    d1,d2
104:          lsl.l    #2,d2
105:          swap.w   d2
106:          move.w   d2,(a2)+
107:
108:          move.w   2(a0),d0
109:          move.w   8(a0),d1
110:          move.w   14(a0),d2
111:          muls.w   (a1)+,d0
112:          muls.w   (a1)+,d1
113:          muls.w   (a1)+,d2
114:          subq.l   #6,a1
115:          add.l    d0,d1
116:          add.l    d1,d2
117:          lsl.l    #2,d2
118:          swap.w   d2
119:          move.w   d2,(a2)+
120:
121:          move.w   4(a0),d0
122:          move.w   10(a0),d1
123:          move.w   16(a0),d2
124:          muls.w   (a1)+,d0
125:          muls.w   (a1)+,d1
126:          muls.w   (a1)+,d2
127:          add.l    d0,d1
128:          add.l    d1,d2
129:          lsl.l    #2,d2
130:          swap.w   d2
131:          move.w   d2,(a2)+
132:
133:          move.w   (a0),d0
134:          move.w   6(a0),d1
135:          move.w   12(a0),d2
136:          muls.w   (a1)+,d0
137:          muls.w   (a1)+,d1
138:          muls.w   (a1)+,d2
139:          subq.l   #6,a1
140:          add.l    d0,d1
141:          add.l    d1,d2
142:          lsl.l    #2,d2
143:          swap.w   d2
144:          move.w   d2,(a2)+
145:
146:          move.w   2(a0),d0
147:          move.w   8(a0),d1
148:          move.w   14(a0),d2
149:          muls.w   (a1)+,d0
150:          muls.w   (a1)+,d1
151:          muls.w   (a1)+,d2
152:          subq.l   #6,a1
153:          add.l    d0,d1
154:          add.l    d1,d2
155:          lsl.l    #2,d2
156:          swap.w   d2
157:          move.w   d2,(a2)+
158:
159:          move.w   4(a0),d0
160:          move.w   10(a0),d1
161:          move.w   16(a0),d2
162:          muls.w   (a1)+,d0
163:          muls.w   (a1)+,d1
164:          muls.w   (a1)+,d2
165:          add.l    d0,d1
166:          add.l    d1,d2
167:          lsl.l    #2,d2
168:          swap.w   d2
169:          move.w   d2,(a2)+
170:
171:          move.b   $80a,d0              *Z座標
172:          move.w   #10000,d2
173:          move.w   #10,d3
174:          btst.l   #5,d0
175:          beq      zzzok
176:          move.w   EZ,d1
177:          add.w    #250,d1
178:          bsr      getd1sub
179:          sub.w    #250,d1
180:          move.w   d1,EZ
181: zzzok:
182:          btst.l   #6,d0
183:          beq      zzz2ok
184:          move.w   EZ,d1
185:          add.w    #250,d1
186:          bsr      getd1add
187:          sub.w    #250,d1
188:          move.w   d1,EZ
189: zzz2ok:
190:          move.w   #-500,d2
191:          move.w   #500,d4
192:
193:          move.w   #-1,d3               *X座標
194:          move.b   $803,d0
195:          btst.l   #6,d0
196:          beq      xxx1ok
197:          move.w   EX,d1
198:          bsr      getd1add2
199:          move.w   d1,EX
200: xxx1ok:
201:          move.w   #1,d3
202:          move.b   $804,d0
203:          btst.l   #0,d0
204:          beq      xxx2ok
205:          move.w   EX,d1
206:          bsr      getd1add2
207:          move.w   d1,EX
208: xxx2ok:
209:          move.w   #-1,d3               *Y座標
210:          move.b   $802,d0
211:          btst.l   #2,d0
212:          beq      yyy1ok
213:          move.w   EY,d1
214:          bsr      getd1add2
215:          move.w   d1,EY
216: yyy1ok:
217:          move.w   #1,d3
218:          move.b   $805,d0
219:          btst.l   #3,d0
220:          beq      yyy2ok
221:          move.w   EY,d1
222:          bsr      getd1add2
223:          move.w   d1,EY
224: yyy2ok:
```

▶うちのX68000XVIはとうとう「RED ZONE」同等品になってしまった。速いぞー。
河田　渉(22)北海道

新製品紹介

ビデオ入力ユニットCZ-6VS1

シャープは新製品として，ビデオ信号をパソコンのデータに変換するビデオ入力ユニット（CZ-6VS1）を発売した。

本機の特徴は，以下のとおり。

1） 映像信号を高画質でパソコンデータに変換

最大1677万色，最高640×480ドットの高解像度でビデオ画像をパソコンのデジタルデータに変換可能（表示は65536色以下)。なお，パソコンの画像をビデオ信号にする機能は持っていない。

2） SCSIの採用

専用スロットを使用せず,SCSI-2(FAST)による接続の採用により，接続可能なパソコンを限定しない汎用化とデータ転送速度の高速化を実現。半面，非SCSI対応機では接続にはSCSIボードが必要となる。また，X68000本体に内蔵された画像取り込み用のハードウェアをまったく使用しないため，従来のようなリアルタイムの取り込みはできなくなっている。

3） 動画や静止画を簡単に保存できるアプリケーションソフトを同梱

MacintoshIIシリーズ用にはQuickTimeをサポートしたツール，X68000ではSX-WINDOW用に動画データ(非圧縮)と画像データ（1677万色は無圧縮TIFFのみ）でセーブできるツールが付属する（X68000版は音声取り込みには非対応)。ここでセーブされた画像はCGAウィンドウ用のデータに変換することもできる。

4） 32ビットCPU：MC68EC020を搭載

画像処理関係のパソコンの負担を軽減している。ビデオ入力ユニットでは常に640×480ドット,1677万色のデータを取り込んでいるらしく，そこからフレームの間引きやデータの拡大/縮小などを32ビットCPUで処理している。それに対し，パソコン側はデータのロード，表示，ファイル作成を担当することになる。

CZ-6VS1 178,000円
シャープ ☎03(3260)1161

画質の向上

ビデオ入力ユニットの最大の利点は取り込みの高画質である。

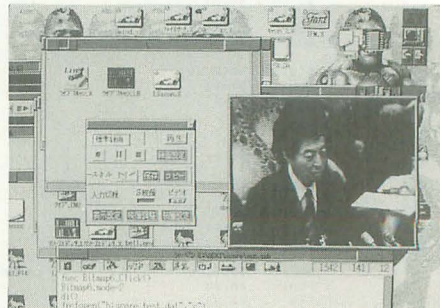
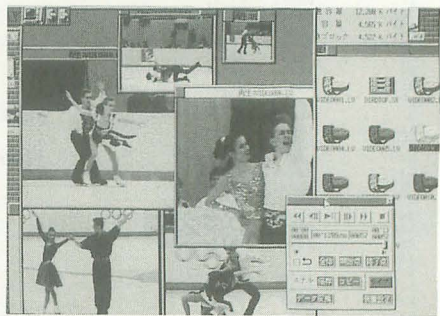
カラーイメージユニットと比べると，まず，画像の左端や上下端に大きく出ていた非映像部のゴミがなくなり，横線が連続している際に出ていた虹色のノイズもやや少なくなったようだ。細かな階調はまだ少し飛びがちだが，色再現性は格段によくなっている。一部の色のザラつき感もなくなった（テレビ地上波をディスプレイテレビCZ-614Dのチューナでモニタリング)。S端子を装備しているので高性能なチューナやビデオデッキなどを使用することでさらに高画質映像を得ることもできるだろう。

ただ，640×480ドットモードの場合，奇数フレームと偶数フレームを合成して走査線数を増やしているのだが，動きの大きいところでは食い違いがかなり気になる。動きの少ない絵では非常に緻密な画像になるだけに残念だ。

それでも全体的に見ると画質の向上は目ざましいといえるだろう。

動画取り込みの基本性能

本機はSX-WINDOW上から専用ツール（ライブスキャン.X）でのみ操作できる。

640×480ドットモードでの取り込みは取り込み表示のみの場合，秒間1コマ程度の表示速度となる。160×120ドットでなら秒間10コマ程度，MOからの再生でも秒間3，4コマが期待できる。データはすべてベタなので，再生速度はデータ量に比例する。ほかのモードは簡単に類推できるだろう。

SX-WINDOW上でテレビを映す

録画した映像を複数同時再生もできる

秒間コマ数では，どの程度の動きなのかわかりづらいと思うが，大雑把にいって，秒間10コマ程度ではややぎこちない感じである。たとえば，会話での口の動きと音声にはかなり違和感がある。

事実上の取り込みレートは，160×120ドットの大きさで秒間10コマが最高と思われる。ちなみに12Mバイト実装のX68030でできるだけ軽いシステムを作り，メモリの許す限り取り込みを行うと(一括モード)，だいたい25秒程度のデータが取り込める（秒間10コマ)。これで約10Mバイトのファイルになる。しかし，再生時はハードディスクから読み込むことになるので秒間10コマでの再生は不可能である。

同様に逐次モードでRAMディスクを使用した場合は秒間8コマ弱が限界のようだ。

本機には，領域未確保のまっさらなハードディスクを用意すれば(FAST SCSI機器にも対応)，それをテンポラリ領域として使うことにより従来の2倍速で録画できる高速録画モードというものも用意されているが，まだ効果は未知数である。

THE SENTINEL

〈対応機種一覧〉●MZ-80 K/C/700/1500●MZ-80 B/2000●MZ-2500/2861●X 1●X 1 turbo/Z●PC-8001/8801/88●SMC-777/C●PASOPIA/5●PASOPIA/7●FM-7/77/AV●MSX/2/2+/turbo R●PC-286/386/486/9801/98/9821●X 68000/X 68030
掲載されたプログラムの利用には各機種用のS-OS"SWORD"システムが必要です。

第143部　S-OSで学ぶZ80マシン語講座(5)

●もうすぐTHE SENTINEL10周年

S-OS"MACE"を機に始まったTHE SENTINELももうすぐ10周年を迎えます。

振り返ってみれば，コンピュータ業界もこの10年でずいぶんと変化してきましたが，THE SENTINELは昔ながらのスタイルで突き進んできました。

しかし，いつまでも自分自身でコンピュータを作り上げる楽しさ，プログラミングの楽しさをなくしたくはありません。

THE SENTINELでも，読者の皆さんと一緒に，64KバイトとZ80の世界でどのようにしたらよりよい環境を手にできるかを追求してきました。全機種共通システムインデックスには，そんな，読者の皆さんの思いが込められている，といったらいいすぎでしょうか。

●アプリケーションのフリーソフト化計画

先ほどもいいましたが，THE SENTINELも10周年を迎えようとしています。しかし，Oh!X(MZ)のバックナンバーに手に入れることのできないものもある現状では，S-OSの古いアプリケーションを手に入れることが事実上不可能となってしまいました。

そこで，せめていままで掲載されてきたアプリケーションのコピー，配布を自由にしようではないか，という考えから「フリーソフト化計画」をぶちあげてみました。

これにより，新しいS-OSユーザーにアプリケーションを譲る，BBSにアップして配布する，ということが公認されます（著作権は作者のまま)。基本的に配布が自由なのはプログラムリストのみで，掲載された記事をコピーして配布することはやめてください（自分でマニュアルを作成したものなら別にかまわない)。

さて，どのようなアプリケーションがコピー配布自由の対象になっているかというと，以下のリストのとおりです。初めての試みということもあり，結構新しめの作品，かつOh!Xスタッフの作品ばかりですが，これからも対象となるアプリケーションを増やしていく予定です。もしも以前THE SENTINELに掲載されたものについて「配布は自由に行っていいよ」という作者の方は，THE SENTINELまでご連絡ください（アンケートハガキで結構です)。特に「SLANG」の制作者である大貫健一さん。また，本誌をお読みでしたらぜひともご連絡ください。

では，ご協力よろしくお願いします。

1986年9月
第28部　FuzzyBASIC
1987年6月
第44部　FuzzyBASICコンパイラ
1989年2月
第77部　高速エディタアセンブラREDA
1989年7月
第82部　TTC用パズルゲームTICBAN
1989年11月
第86部　TTI用パズルゲームPUSH BON!
1990年1月号
第88部　SLANG用ゲームWORM KUN
1990年6月号
第94部　STACK用ゲームSQUASH!
1990年7月号
第96部　リロケータブルアセンブラWZD
1990年8月号
第97部　リンカWLK
1990年11月号
第100部　タブコード対応エディタEDC-T
1991年3月号
第104部　アクションゲームMAD BALLIN'
1991年9月号
第110部　SLANG用NEWファイル入出力ライブラリ
1991年11月号
第113部　MORTAL
1992年2月号
第116部　シミュレーションゲームPOLANYI
1992年4，6，8月号 実践Small-C講座
第118部　オプティマイザO80
第120部　COMMAND.OBJ
第122部　ワイルドカード
1992年10月号
第125部　SLENDER HUL
1992年11月号
第126部　EDIT
1993年6月号
第133部　REVERSI
1993年7月号
特別付録 MSX用S-OS"SWORD"
1993年9月号
第135部　7並べ

1994■インデックス

全機種共通
S-OS"SWORD"要

S-OSで学ぶ
Z80マシン語講座(5)

Itou Masahiko
伊藤 雅彦

プログラム制作も順調に進み，この連載も残すところあと1回。だいぶ大掛かりなプログラムになってきましたが，リストに圧倒されずアルゴリズムをよく学んでください。

ディスクエディタの制作が着々と進んでいますが，それとともにソースリストもだんだんと大きくなってきました。リストをコメントまで真面目に入力している人は，もしかしたらすでにアセンブルができなくなっているかもしれませんね。アセンブルしたらわけのわからないエラーが出て，途方に暮れていたという人がいたらごめんなさい。ソースリストが大きくなってきたときの対処法を囲みで説明しておきますので，参照してください。

Wコマンド

今月はセクタの内容を書き換えるWコマンドからいきましょう。Wコマンドはいままで作ってきたコマンドと大きく違うところがあります。いままでのものはコマンドが入力されたらそれに応じた処理をしてすぐにコマンド入力に戻っていましたが，Wコマンドではセクタのデータをエディットするという処理が必要になります。ですから，Wコマンドが入力されるとコマンド入力モードからデータエディットモードに入るという形になります。

このデータエディットをどういう仕様のものにするか，操作性に大きくかかわる部分だけに悩むところです。セクタ書き換えコマンドは，ディスクエディタにとって必須のコマンドなので，本当はもっと早く作るべきだったのですが，仕様を考えていたら遅くなってしまいました。

さて，エディット処理の方法には大きく分けて2つの方法が考えられます。1文字のキー入力があるたびにデータを更新する方法と，1行入力されて初めてデータ更新処理をする方法です。テキストエディタでいえば，スクリーンエディタとカーソルエディタ(またはラインエディタ)の違いです。

私としては，スクリーンエディットのほうが好みです。カーソルエディットでは，リターンキーを押さないと画面上では書き換えたはずのデータが内部では書き換えられていないということが起きてしまいます。パソコンを始めたばかりのときに，BASICのカーソルエディタでプログラムを入力している際に行の最後にリターンキーを押すのを忘れて，全然記憶されていなかったという経験をされた人もいるんじゃないでしょうか。画面に表示されているものと実際のデータが違うのは，ちょっと気持ち悪い感じがします。

ですが，S-OSでプログラムを組もうとすると，カーソルエディットのほうがとっても楽です。それは，1行入力ルーチン#GETLが使えるからです。#GETLではカーソル移動や文字削除，挿入など，各機種用の編集操作ができますので，これを使うだけでそこそこの操作性のエディタを簡単に作ることができます。スクリーンエディットをやろうと思ったら，カーソル移動や画面表示の制御をしながら，いろいろな編集機能を自前で実現しなくてはいけませんから大変です。

というわけで，今回はカーソルエディットでいくことにしました。ただし，画面表示と実際のデータをできるだけ一致させるために，1行入力されたら画面のほぼ全体を書き直す処理をすることにします。早い話が，MACINTO-Cと同じようなものになるわけです。こうすると，リターンキーを押したあと，画面を書き直すわずかの間だけキー入力を受けつけなくなってしまいますが，ま，ちょっとだけですから（機種にもよるでしょうが）いいでしょう。

では，Wコマンド処理を組み立てていきましょう。大まかに捉えると，

1) ディスクから1セクタ読み込む
2) セクタデータをエディット
3) データをディスクへ書き込む

ということになります。このうち1)のディスク読み込みについては，Rコマンドと似たようなパラメータ処理をして#DRDSBをコールすればいいため，簡単に実現できます。3) のディスク書き込みも#DWTSBをコールするだけです。となると，やはり問題はエディット部分ということになります。

エディット部分の処理手順を細かく考えてみると，

1) セクタデータを画面に表示
2) 1行入力
3) 入力された内容に従ってデータを書き換える
4) 1)に戻る

となります。1)の表示処理はRコマンドと同様にDUMPルーチンを使えばいいし，2)の1行入力は#GETLルーチンをコールするだけ。つまり，問題は3)の部分ですね。

今月号で使用したシステムサブルーチン&ワークエリア

●サブルーチン
#BRKEY：BREAKキーが押されているかどうかをZフラグにセット
#LOC　：カーソル位置を(L,H)に移動
●ワークエリア
#MEMAX：メモリのフリーエリアの上限

入力された行の解釈をどうやるかをはっきり決めておく必要があります。

まず，入力される行というのはダンプ表示の行ですから，

1,2桁目：オフセット値
3桁目　：コロン（「：」）
4桁目～：データ

という形式になっています。そこで，

1) 1,2桁目を2桁の16進数とみなしてオフセット値を得る
→ 16進数でなければ入力無効
2) 3桁目がコロンかどうかチェック
→ コロンでなければ入力無効
3) 4桁目以降をデータを表す文字列とみなしてデータ列を得る
4) オフセット値の指すアドレス以降のデータを3) で得たデータ列に書き換える

という処理をすることにします。

いまのところ，まだ3)のところが明確になっていませんね。基本的に4桁目以降にはデータが2桁の16進数の形で並んでいるため，それを順に拾っていくという考え方でプログラムを組めば大丈夫です。

しかし，ここでちょっと機能を追加したいという気がします。大したものではなくて，むしろ当たり前の機能なんですが，データを2桁の16進数のほかにキャラクタ文字列でも入力できたら，使いやすくなりそうですよね。というわけで，クォーテーション（「'」または「"」）で囲んだ文字列があったら，そのキャラクタコードをデータとすることにします。たとえば，

'S-OS'　→ 53_H $2D_H$ $4F_H$ 53_H
"MSG"0D → $4D_H$ 53_H 47_H $0D_H$

といった具合です。クォーテーションを2種類使えるようにしたのは，クォーテーション自身を文字列に入れたいときに，2種類あれば「" ' "」とか「' " '」と指定できるからです。

この機能を実現するためにどうしたらいいか説明していきましょう。入力された文字列を解釈してデータ列を作るとき，最初は2桁の16進数が書かれているものとして前から順にデータを生成していきます。そして，クォーテーションを見つけたら，今度はそれ以降の文字をそのまま（キャラクタコードとして）データにしていきます。再び前と同じクォーテーションを見つけたら，2桁の16進数とみなしてデータを生成する処理に戻ります。つまりクォーテーションがデータの解釈方法を変えるスイッチになるわけですね。

では，実際に作ったプログラムを見てみましょう。ディスクの読み込みや書き込みの部分は省略して，データエディット処理だけ説明します。WCOM2からWCOM11の手前までがデータエディット処理のループになります。

まず，セクタデータを画面に表示する処理をしていますが，ここを作っていてハタと気づいたのが，40桁画面モードでは1画面で全部のデータを表示しきれないということです。結局データの前半と後半を切り替えて表示することで解決しました。データ入力のときに4桁目（コロンのすぐあと）に「X」を入力すると，表示するデータが切り替わるようにしています。前後半どちらを表示するかはA'レジスタに保持しています（00_Hなら前半，FF_Hなら後半）。

そして，意外に面倒だったのが#GETLをコールする前のカーソル位置の制御でした。カーソルは基本的には入力した行の次の行に出てほしいものです。WCOM7以降でカーソルを出す行をAレジスタに求めていますが，やっていることは，入力された行のオフセット値をAレジスタに入れ，

●80桁モードの場合
1) A←A/16+1
2) A>15なら，A←15
●40桁モードで表示データが前半の場合
1) A←A/8+1
2) A>15なら，A←15
●40桁モードで表示データが後半の場合
1) A←A/8−16
2) A<0なら，A=0
　それ以外なら，
3) 　A=A+1
4) A>15なら，A←15

という処理です。こうして，データが表示されている最初の行を0行目とした場合の行の番号を求めて，ループの先頭に戻ります。あとは#GETLをコールする前に（3，A+2）にカーソルをもっていくことで，カーソル位置の制御をしています。

また，4桁目以降のデータ列を解釈する処理はサブルーチンGETSTRでやっています。サブルーチンにしたのは，Sコマンドの処理で検索データ列のパラメータを解釈するときにも使おうという思惑があるからです。このルーチンではワークエリアSTRINGにデータ列を作ります。このワークの先頭からCレジスタの文字数分のデータ列が，有効なデータ列になります。

行頭のオフセット値を得るときやデータ列の解釈のときには，2桁の16進数を数値に変換するGET2HEXルーチンを使っています。これはシステムサブルーチンの#2HEXとほぼ同じものです。#2HEXでは小文字表記の16進数をエラーとしてしまうのが気に入らなかったので，自前で用意しました（これは余計なこだわりだったかな）。

おっと，エディット処理を抜けるときの操作をいい忘れていました。まずSHIFT+BREAKキーでとりあえずエディットモードを抜けます。ここでリターンキーを押せばエディットした内容をディスクに書き込んで処理終了，SHIFT+BREAKキーなら書き込まずに終了，スペースキーならエディットモードに戻ります。

Sコマンド

ではSコマンド，データ列の検索にいきます。第1，第2パラメータで検索対象になる範囲の先頭，最終レコード番号を指定し，第3パラメータで検索データ列を指定

プログラムが大きくなったら

プログラムが大きくなってメモリにソースとオブジェクトを置く余裕がなくなったときには，2つの対処方法があります（一応断っておくと，これはREDAおよびZEDAでの話です）。

1) ソースを破壊しながらアセンブル

オブジェクト生成開始アドレスをソースの格納先頭アドレスと同じにします。つまり，ソースが5000_Hから格納されているならOFFSET疑似命令で，

OFFSET 05000H-03000H

とするのです。こうすれば，ソースがあったところにオブジェクトコードを上書きしていきますから，ソースリストが読み込めるだけのメモリがあればいいわけです。こんなことをしていいのかと思われるかもしれませんが，「オブジェクトの大きさは，それを生成するソースの大きさより小さい」「オブジェクトはソースを前から順に解析しながら，前から順に生成される」という理由で，大丈夫なのです。ただし，DEFS疑似命令を使ったときは第1の理由を満たさないことがありますから注意してください。

2) 分割アセンブル・断片アセンブル

ソースのファイルを分割することで，ソース読み込み領域を小さく抑える方法です。詳しくはアセンブラ掲載号（1993年12月号）の説明を見ていただくとして，ここでは一番簡単なREDAの断片アセンブルの方法を説明しましょう。

まずソースファイルを分割します。ADDIEならコマンド処理部分までと基本サブルーチン以降の2つに分けるのがいいでしょう。それぞれを，たとえばADDIE1，ADDIE2というファイル名でセーブします。

そしてREDAを起動して，

/F
A ADDIE1：ADDIE2

と打ち込んでアセンブル完了です。

します。検索データ列の書式はWコマンドのときと同じで，2桁の16進数かクォーテーションで囲んだ文字列で指定できます。

この文字列検索の処理はどうやったらいいのでしょう。本誌を丹念に読んでいる人はすでに知っているはずです。そうでなくても，真面目に読んでいる人なら1993年3月号の「X68000マシン語プログラミング」の記事を探し当てていることでしょう。そこに文字列検索のアルゴリズムが親切に解説されています。ありがたいことです。ありがたく勉強させていただいて，そこで説明されているボイヤー・ムーア法（BM法）というのを使うことにしましょう。

BM法の詳しい解説は参照記事を見ていただくとして，ここではアルゴリズムをざっと説明します（図1）。

まず256バイトのテーブルを用意して，00_H～FF_Hのデータが検索データ列の末尾の位置から何バイト前にあるかを登録しておきます。検索データ列の中にない場合は検索データ列の文字数を登録します。たとえば検索データ列が，

03_H 00_H 03_H 02_H

なら，テーブルは，

2 4 0 1 4 4 …（以後すべて4）

となります。

そして，検索データ列と検索対象データ列を，検索データ列の末尾から比較していきます。末尾が一致していたら前へ順に比較を進めます。不一致になったら，その不一致箇所の検索対象データ列側のデータをキーにして，最初に作ったテーブルを参照します。そしてテーブルから得た値の分だけ比較位置を進めて，そこから再び検索データ列の末尾との比較を始めます。ただし，そのときに比較位置が前に末尾と比較した位置より前に戻ってしまうときには，前の比較位置より2バイト先に進めるようにします。あっさりしすぎた説明ではありますが，以上です。

これで検索自体はOKです。でも，検索対象データはメモリ上に置く必要があります。ところが，2Dディスク全体を検索対象範囲にした場合には，その範囲をいっぺんにメモリに読み込むことはできませんよね。そこで何回かに分けて読み込んで検索を行うことになるんですが，そのとき注意しなくてはいけないのが，データ列の分かれ目の部分です。

たとえば，「12345678」から「345」を検索するとして，一度に4バイトだけメモリに読み込めるとしましょう。ここで単純に「1234」と「5678」に分けてそれぞれを検索したら，「345」は見つかりません（図2-a）。これではまずいので，読み込み領域の前に少し余裕をもたせて1回目に読み込んだ「34」を「5678」の前にくっつけて検索するようにします（図2-b）。こうすると「345」が無事検索されます。

アルゴリズムがわかったところで，プログラムを見ましょう。まずワークエリアの取り方ですが，検索対象データ列を一度になるべくたくさん読みたいので，できるだけ大きなワークエリアがほしいところです。そこで，先月号でコマンド処理用の多目的ワークとして確保したCOMWKの大きさを拡張することにします。先月号は4Kバ

レジスタペアのゼロチェック

レジスタペアの値が0かどうか検査するときの常套手段が，

```
LD  A,H
OR  L
```

です（HLレジスタペアの場合）。これでZフラグが1ならHL＝0だとわかります。

この方法は，2バイト値をカウンタにしてループ処理をするときに便利に使えます。レジスタペアのDEC命令はZフラグが不変なので，ループを組むときには困ってしまうのですが，

```
        LD   HL,カウンタ値
LOOP:
        (ループ内処理)
        DEC  HL    : Zフラグは変化しない
        LD   A,H
        OR   L
        JR   NZ,LOOP
```

とすればOKというわけです。

図1 BM法の検索アルゴリズム

```
0 1 3 2 3 2 3 0 3 2 1
から
3 0 3 2
を探すとする。
テーブル：| 0 | 1 | 2 | 3 |
          | 2 | 4 | 0 | 1 |

(1) 3 0 3 2
    0 1 3 2 3 2 3 0 3 2 1
          ↑
    ここから照合開始

(2) 3 0 3 2
    0 [1] ③ ② 3 2 3 0 3 2 1
      └──────┘↑
    4バイト進める

(3)     3 0 3 2
    0 1 3 [2] ③ ② 3 0 3 2 1
              ↓
    0バイト進める＝進めない

(4) 3 0 3 2
    0 1 3 2 3 2 3 0 3 2 1
          ↑
    戻っしまうので駄目

(5)         3 0 3 2
    0 1 3 2 3 2 3 0 3 2 1
            └──┘↑
    2バイト進める

(6)           3 0 3 2
    0 1 3 2 3 2 3 [0] 3 2 1
                  └──┘↑
    2バイト進める

(7)               3 0 3 2
    0 1 3 2 3 2 ③ ⓪ ③ ② 1
                  ↑
    照合成功

(8)                   3 0 3 2
    0 1 3 2 3 2 3 0 3 2 1
                          ↑
    次から照合再開
```

図2 分割読み込みしながらの検索

```
a．失敗例

1 2 3 4 5 6 7 8
└──────┘
   ↘
| 1 | 2 | 3 | 4 | 「『3 4 5』はないなあ」

1 2 3 4 5 6 7 8
        └──────┘
           ↙
| 5 | 6 | 7 | 8 | 「こっちにもない」

b．成功例

1 2 3 4 5 6 7 8
└──────┘
   ↘
|   |   | 1 | 2 | 3 | 4 | 「『3 4 5』はないなあ」
        ↓
| 3 | 4 | 1 | 2 | 3 | 4 | 「後ろのほうを前に移して，と」

1 2 3 4 5 6 7 8
        └──────┘
           ↙
| 3 | 4 | 5 | 6 | 7 | 8 | 「あった！」
```

イト確保していましたが，これをメモリが許す限り大きくしてしまいましょう。どうするかというと，とりあえず先月号の，

COMWK：

DS 4096

という部分を消して，リストの最後に，

COMWK：

と書きます。これでラベルCOMWKにはプログラムの最終アドレスが定義されますから，このアドレス以後フリーエリアの最後までをCOMWKのワークエリアとして使えるわけです（なお，プログラムをセーブするときにフリーエリアの最後までセーブする必要はありません。このプログラムではどのワークにも初期値がありませんから，ワークエリアの部分はセーブしなくていいのです)。ちなみにフリーエリアの最終アドレスは#MEMAXを見ればわかります。

そして，COMWKの先頭256バイトをBM法で必要なテーブル領域として使い，それ以後を検索対象データ列の読み込み領域にしています。ただし最初の256バイトは，先ほど説明した，前回読み込んだデータ列の末尾を入れておく「余裕」としています。

処理の流れのほうはリストのコメントで理解していただくことにします。ちょっと仕様の補足をしますと，検索データ列が見つかった場合は，その場所のレコード番号とセクタ先頭からのオフセット値を「00A1-7D」といった形式で表示します。

また，表示直後にスペースキーでポーズが効き，メモリに読み込んだデータ列を検索し終わるたびにSHIFT＋BREAKキーによる処理の中止ができるようになっています。

＊　＊　＊

いよいよ来月号でプログラムは完成です。つまり最終回ってことです。もうひと息がんばりましょう。

リスト

```
  1: ;
  2: ; ラベル定義
  3: ;
  4: #BRKEY: EQU 01FCDH
  5: #LOC:   EQU 0201EH
  6: ;
  7: #MEMAX: EQU 01F6AH
  8: ;
  9: ; コマンド処理ルーチン
 10: ;
 11: ;
 12: ; W Command
 13: ;
 14: WCOM:
 15:  CALL PARAMETER
 16:  JP   C,#BELL
 17:  JR   NZ,WCOM1
 18:  LD   HL,(RECORD)
 19:  LD   A,(WRCBK)
 20:  OR   A
 21:  JR   Z,WCOM1
 22:  DEC  HL
 23: ;
 24: WCOM1:
 25:  CALL #MPRNT
 26:  DB   00CH         ; 画面消去
 27:  DM   'Record '
 28:  DB   0
 29:  CALL #PRTHL
 30:  CALL #LTNL
 31: ;
 32:  LD   A,(DEVICE)
 33:  LD   (#DSK),A
 34:  EX   DE,HL
 35:  LD   HL,(#DTBUF)
 36:  LD   A,1
 37:  CALL #DRDSB
 38:  JP   C,#ERROR
 39: ;
 40:  PUSH DE           ; レコード番号を保護
 41:  INC  DE
 42:  LD   (RECORD),DE
 43:  XOR  A
 44:  LD   (WRCBK),A
 45: ;
 46:  CALL SCALE
 47:  XOR  A
 48:  EX   AF,AF'       ; A'= 0 (40桁モードで前半を表示)
 49:  XOR  A            ; カーソル位置は0行目
 50: ;
 51: WCOM2:
 52:  PUSH AF          ; カーソル位置を保護
 53:  LD   HL,00200H ; H = 2, L = 0
 54:  CALL #LOC
 55:  LD   HL,(#DTBUF)
 56:  LD   A,(WIDMODE)
 57:  DEC  A
 58:  JR   Z,WCOM3     ; 80桁モードなら WCOM3 へ
 59:  EX   AF,AF'
 60:  LD   C,A
 61:  EX   AF,AF'
 62:  XOR  A
 63:  INC  C
 64:  JR   NZ,WCOM3    ; 前半を表示するなら WCOM3 へ
 65:  LD   DE,080H
 66:  ADD  HL,DE
 67:  LD   A,080H
 68: WCOM3:
 69:  LD   (DUMPOS),A
 70:  LD   C,16
 71: WCOM4:
 72:  CALL DUMP         ; セクタデータ表示
 73:  DEC  C
 74:  JR   NZ,WCOM4
 75:  POP  AF
 76: ;
 77:  ADD  A,2
 78:  LD   H,A
 79:  LD   L,3
 80:  CALL #LOC         ; カーソルを (3,A+2) へ
 81:  LD   DE,(#KBFAD)
 82:  CALL #GETL
 83: ;
 84:  LD   A,(DE)
 85:  CP   01BH
 86:  JR   Z,WCOM11    ; SHIFT+BREAK チェック
 87:  CALL GET2HEX     ; オフセット値を取得
 88:  JR   C,WCOM10
 89:  LD   H,A
 90:  LD   A,(DE)
 91:  CP   ':'
 92:  JR   NZ,WCOM10 ; コロンチェック
 93:  INC  DE
 94:  LD   A,(DE)
 95:  CALL CAPITAL
 96:  CP   'X'         ; 前・後半表示切り替えか
 97:  JR   NZ,WCOM5
 98:  LD   A,(WIDMODE)
 99:  OR   A
100:  JR   NZ,WCOM5
101:  EX   AF,AF'
102:  CPL               ; Aレジスタ反転
103:  EX   AF,AF'
104:  XOR  A            ; カーソル位置は0行目
105:  JR   WCOM2
106: ;
107: WCOM5:
108:  PUSH HL           ; オフセット値を保護
109:  CALL GETSTR       ; 変更データ列を取得
110:  INC  C
111:  DEC  C
112:  JR   Z,WCOM7     ; 0文字なら WCOM7 へ
113:  POP  AF
114:  PUSH AF
115:  NEG               ; A = 256 - オフセット値
116:  JR   Z,WCOM6
117:  CP   C
118:  JR   NC,WCOM6
119:  LD   C,A          ; 256バイト目を超えるデータは無視する
120: WCOM6:
121:  LD   B,0          ; BC = 変更データ長
122:  NEG
123:  LD   E,A
124:  LD   D,B          ; DE = オフセット値
125:  LD   HL,(#DTBUF)
126:  ADD  HL,DE
127:  EX   DE,HL
128:  LD   HL,STRING
129:  LDIR              ; データ変更
130: ;
131: WCOM7:
132:  LD   A,(WIDMODE)
133:  OR   A
134:  JR   Z,WCOM8
135:  POP  AF
136:  AND  0F0H
137:  RRCA
138:  RRCA
139:  RRCA
140:  RRCA              ; A = A / 16
141:  JR   WCOM9
142: WCOM8:
143:  POP  AF
144:  AND  0F8H
145:  RRCA
146:  RRCA
147:  RRCA              ; A = A / 8
148:  EX   AF,AF'
149:  LD   H,A
150:  EX   AF,AF'
151:  INC  H
152:  JR   NZ,WCOM9
153:  SUB  010H
154:  JR   C,WCOM10
155: ;
156: WCOM9:
157:  INC  A
158:  CP   010H
159:  JP   C,WCOM2
160:  LD   A,00FH
```

```
161:  JP    WCOM2
162:  ;
163: WCOM10:
164:  XOR   A
165:  JP    WCOM2
166:  ;
167: WCOM11:
168:  LD    HL,01300H ; H = 19 , L = 0
169: WCOM12:
170:  CALL  #LOC
171:  CALL  #MPRNT
172:  DM    'write=[RET],break=[BRK],re-edit=[SPC] '
173:  DB    0
174:  CALL  #FLGET
175:  LD    C,A
176:  CALL  #LOC
177:  LD    B,37
178: WCOM13:
179:  CALL  #PRNTS
180:  DJNZ  WCOM13
181:  LD    A,C
182:  SUB   ' '           ; 「CP」でなく「SUB」なのは、条件成立時に
183:  JP    Z,WCOM2       ;   A=0 でジャンプしたいから
184:  CP    0DH-' '
185:  JR    Z,WCOM14
186:  CP    01BH-' '
187:  JR    NZ,WCOM12
188:  ;
189:  CALL  #LTNL
190:  POP   DE            ; 保護していたオフセット値を破棄
191:  RET
192:  ;
193: WCOM14:
194:  CALL  #LTNL
195:  POP   DE
196:  LD    A,(DEVICE)
197:  LD    (#DSK),A
198:  LD    HL,(#DTBUF)
199:  LD    A,1
200:  CALL  #DWTSB
201:  JP    C,#ERROR
202:  RET
203:  ;
204:  ; S Command
205:  ;
206: SCOM:
207:  CALL  PARAMETER
208:  JP    C,#BELL
209:  JP    Z,#BELL
210:  LD    C,L
211:  LD    B,H           ; BC = 検索開始レコード番号
212:  CALL  PARAMETER ; HL = 検索終了レコード番号
213:  JP    C,#BELL
214:  JP    Z,#BELL
215:  EXX
216:  LD    DE,(KBPTR)
217:  CALL  GETSTR    ; 検索データ列を取得
218:  JP    C,#BELL
219:  LD    A,C           ; A = 検索データ長
220:  OR    A             ; (ここで必ず Cy=0 になる)
221:  JP    Z,#BELL
222:  EXX
223:  SBC   HL,BC     ; HL = HL - BC
224:  JP    C,#BELL
225:  INC   HL        ; HL = 検索対象セクタ数
226:  ;
227:  EXX
228:  LD    (SCOMSL),A
229:  LD    E,A
230:  DEC   E
231:  LD    D,0
232:  LD    HL,STRING
233:  ADD   HL,DE
234:  LD    (SCOMSE),HL
235:  LD    HL,COMWK
236:  LD    DE,COMWK+1
237:  LD    BC,255
238:  LD    (HL),A
239:  LDIR                ; テーブルを検索データ長で埋める
240:  LD    HL,STRING
241:  LD    BC,COMWK
242:  PUSH  AF            ; 検索データ長を保護
243: SCOM1:               ; テーブル作成ループ
244:  LD    E,(HL)
245:  INC   HL
246:  LD    D,0
247:  EX    DE,HL
248:  ADD   HL,BC
249:  EX    DE,HL     ; DE = DE + BC
250:  DEC   A
251:  LD    (DE),A
252:  JR    NZ,SCOM1
253:  POP   AF
254:  ;
255:  LD    DE,(#MEMAX)
256:  LD    HL,253
257:  ADD   HL,DE
258:  JR    NC,SCOM2
259:  LD    DE,0FF02H
260: SCOM2:
261:  LD    HL,-COMWK-512
262:  ADD   HL,DE     ; H = 1度に読み込めるセクタ数
263:  LD    A,H
264:  EXX
265:  LD    E,A
266:  LD    D,0
267:  ;
268:  CALL  SCOMS1
269:  JP    C,#ERROR
270:  LD    HL,COMWK+512-1
271:  LD    A,(SCOMSL)
272:  LD    E,A
273:  LD    D,0
274:  ADD   HL,DE
275:  EX    DE,HL           ; DE = 照合開始位置
276:  CALL  PRTRSET
277:  ;                           ここからループ
278: SCOM3:
279:  LD    B,0
280: SCOM4:
281:  LD    A,(DE)          ; A = 検索データ列の末尾と照合されるデータ
282:  LD    HL,COMWK
283:  LD    C,A
284:  ADD   HL,BC
285:  LD    A,(HL)          ; テーブルを参照
286:  OR    A
287:  JR    Z,SCOM7         ; 0なら末尾は一致しているので SCOM7 へ
288:  LD    C,A
289:  EX    DE,HL
290:  ADD   HL,BC
291:  EX    DE,HL           ; 照合位置を進める
292: SCOM5:
293:  LD    HL,(SCOMEC)
294:  ADD   HL,DE           ; 検索終了か
295:  JP    NC,SCOM4
296:  ;
297:  CALL  #BRKEY
298:  JP    Z,#LTNL
299:  EXX
300:  LD    A,H
301:  OR    L               ; (ここで必ず Cy=0 になる)
302:  JP    Z,#LTNL         ; HL=0 (未処理セクタがない) なら処理完了
303:  LD    A,(SCOMSL)
304:  DEC   A
305:  JR    Z,SCOM6
306:  EXX
307:  LD    C,A
308:  LD    B,0
309:  LD    DE,(SCOMEC)
310:  LD    HL,-1
311:  SBC   HL,DE           ; HL = HL - DE
312:  LD    DE,COMWK+512-1
313:  LDDR                  ; 検索対象データ列の末尾 (検索データ長
314:                        ; −1バイト分) を前へ移動
315:  EXX
316: SCOM6:
317:  LD    A,E
318:  EX    DE,HL
319:  ADD   HL,BC
320:  EX    DE,HL           ; DE = DE + BC
321:  LD    C,E
322:  LD    B,D             ; 読み込み先頭セクタを更新
323:  LD    E,A
324:  LD    D,0
325:  CALL  SCOMS1
326:  JP    C,#ERROR
327:  LD    DE,COMWK+512
328:  JR    SCOM3
329:  ;
330: SCOM7:
331:  LD    A,(SCOMSL)
332:  LD    C,A
333:  LD    HL,(SCOMSE)
334:  PUSH  DE              ; 末尾照合位置を保護
335: SCOM8:
336:  DEC   C               ; C = 照合残りバイト数
337:  JR    Z,SCOM9
338:  DEC   DE
339:  DEC   HL
340:  LD    A,(DE)
341:  CP    (HL)            ; 照合
342:  JR    Z,SCOM8
343:  ;
344:  LD    HL,COMWK
345:  LD    C,A
346:  ADD   HL,BC
347:  LD    C,(HL)          ; テーブルを参照
348:  EX    DE,HL
349:  ADD   HL,BC           ; (ここで必ず Cy=0 になる)
350:  EX    DE,HL           ; DE = DE + BC
351:  POP   HL
352:  SBC   HL,DE           ; HL = HL - DE (バックスライドするか)
353:  JP    C,SCOM5
354:  ADD   HL,DE
355:  EX    DE,HL
356:  INC   DE
357:  INC   DE              ; 2バイト進める
358:  JP    SCOM5
359:  ;                       照合成功
360: SCOM9:
361:  LD    HL,-COMWK-512
362:  ADD   HL,DE
363:  LD    C,L
364:  LD    L,H
365:  SBC   A,A             ; Cy=0 なら A=0 , Cy=1 なら A=FFH
366:  CPL
367:  LD    H,A             ; HL = 読み込み先頭セクタからのオフセット
368:  EXX
369:  PUSH  BC
370:  EXX
371:  POP   DE
372:  ADD   HL,DE
373:  CALL  #PRTHL
374:  LD    A,'-'
375:  CALL  #PRINT
376:  LD    A,C
377:  CALL  #PRTHX
378:  CALL  #PRNTS
379:  POP   DE
380:  CALL  #PAUSE
381:  DW    #LTNL
382:  INC   DE
```

```
383:  JP   SCOM5
384: ;
385: ; ディスク読み込み&裏レジスタセット& SCOMEC セット
386: ;   in ---- BC = 読み込み先頭レコード番号
387: ;           DE = 読み込みセクタ数
388: ;           HL = 未処理セクタ数
389: ;   out --- Cy = 読み込み失敗(1)／成功(0)
390: ;           BC'= 読み込み先頭レコード番号
391: ;           DE'= 読み込みセクタ数
392: ;           HL'= 未処理セクタ数
393: ;   break - F, A, BC, DE, HL, F',A'
394: ;
395: SCOMS1:
396:  OR   A          ; Cy=0
397:  SBC  HL,DE
398:  JR   NC,SCOMS11
399:  ADD  HL,DE
400:  EX   DE,HL
401:  LD   HL,0
402: SCOMS11:
403:  PUSH HL
404:  PUSH DE
405:  LD   A,(DEVICE)
406:  LD   (#DSK),A
407:  LD   A,E
408:  LD   HL,COMWK+512
409:  LD   E,C
410:  LD   D,B
411:  CALL #DRDSB
412:  POP  DE
413:  POP  HL
414:  RET  C
415: ;
416:  LD   A,E
417:  EXX
418:  ADD  A,COMWK/256+2 ; COMWK の上位バイト+2を加算
419:  CPL
420:  LD   H,A
421:  LD   L,-COMWK-1 ; COMWK の下位バイトをビット反転
422:                  ; した値を代入
423:  INC  HL         ; HL = 10000H - (COMWK+512+E*256)
424:  LD   (SCOMEC),HL
425:  OR   A          ; Cy=0
426:  RET
427: ;
428: ; 基本サブルーチン群
429: ;
430: ;
431: ; GETSTR
432: ;    STRING にデータ列を取得
433: ;
434: ;   in ---- DE = データ列指定データ先頭アドレス
435: ;   out --- Cy = 書式エラー(1)／正常(0)
436: ;           C  = データ長
437: ;   break - F, A, B, DE, HL
438: ;
439: GETSTR:
440:  LD   HL,STRING
441:  LD   C,0
442: GETSTR1:
443:  LD   A,(DE)
444:  INC  DE
445:  CP   ' '
446:  JR   Z,GETSTR1 ; スペースをカット
447:  OR   A
448:  RET  Z          ; データエンド
449:  CP   '''
450:  JR   Z,GETSTR2
451:  CP   '"'
452:  JR   Z,GETSTR2
453:  DEC  DE
454:  CALL GET2HEX
455:  RET  C
456:  LD   (HL),A
457:  INC  HL
458:  INC  C
459:  JR   GETSTR1
460: ;
461: GETSTR2:
462:  PUSH BC          ; データ長を保護
463:  LD   B,A
464:  JR   GETSTR4
465: GETSTR3:
466:  LD   (HL),A
467:  INC  HL
468:  INC  C
469: GETSTR4:
470:  LD   A,(DE)
471:  INC  DE
472:  OR   A
473:  JR   Z,GETSTR5 ; 対応するクォーテーションかなかった
474:  CP   B
475:  JR   NZ,GETSTR3
476:  POP  AF          ; データ破棄
477:  JR   GETSTR1
478: GETSTR5:
479:  POP  BC
480:  SCF
481:  RET
482: ;
483: ; GET2HEX
484: ;     2桁16進数 → 数値変換
485: ;
486: ;   in ---- DE = 16進数データ先頭アドレス
487: ;   out --- Cy = 変換失敗(1)／成功(0)
488: ;           A  = 数値 (Cy=0 の時)
489: ;           DE = DE + 2 (Cy=0 の時)
490: ;   break - F, B
491: ;
492: GET2HEX:
493:  LD   A,(DE)
494:  CALL CAPITAL
495:  CALL #HEX
496:  RET  C
497:  ADD  A,A
498:  ADD  A,A
499:  ADD  A,A
500:  ADD  A,A          ; A = A * 16
501:  LD   B,A
502:  INC  DE
503:  LD   A,(DE)
504:  CALL CAPITAL
505:  CALL #HEX
506:  RET  C
507:  OR   B
508:  INC  DE
509:  RET
510: ;
511: ; ワークエリア
512: ;
513: STRING:          ; 文字列格納領域
514:  DS 254
515: SCOMSL:          ; 検索データ長
516:  DS 1
517: SCOMSE:          ; 検索データ末尾アドレス
518:  DS 2
519: SCOMEC:          ; 検索終了チェック用
520:  DS 2
521: ;
522: ; COMWK の変更
523: ;
524: ;   先月追加した
525: ;     COMWK:
526: ;      DS 4096
527: ;   を削除して、リストの最後に
528: ;     COMWK:
529: ;   と書いてください。
530:
```

▶全機種共通システムインデックス◀

＊以下のアプリケーションは，基本システムであるS-OS "MACE" またはS-OS "SWORD" がないと動作しませんのでご注意ください。

▶X68000を購入して約半年，某国民機はホコリをかぶっています。

渕 秀嗣(21)鹿児島県

第22部　magiFORTH TRACER
第23部　ディスクダンプ＆エディタ
第24部　“SWORD” 2000 QD
連載　対話で学ぶmagiFORTH
特別付録　PC-8801版S-OS “SWORD”
■86年7月号
第25部　FM音源ミュージックシステム
付録　FM音源ボードの製作
連載　計算力アップのmagiFORTH
特別付録　SMC-777版S-OS “SWORD”
■86年8月号
第26部　対局五目並べ
第27部　MZ-2500版S-OS “SWORD”
■86年9月号
第28部　FuzzyBASIC発表
連載　明日に向かってmagiFORTH
■86年10月号
第29部　ちょっと便利な拡張プログラム
第30部　ディスクモニタDREAM
第31部　FuzzyBASIC料理法<1>
■86年11月号
第32部　パズルゲームHOTTAN
第33部　MAZE in MAZE
連載　FuzzyBASIC料理法<2>
■86年12月号
第34部　CASL & COMET
連載　FuzzyBASIC料理法<3>

1987

■87年1月号
第35部　マシン語入力ツールMACINTO-C
連載　FuzzyBASIC料理法<4>
■87年2月号
第36部　アドベンチャーゲームMARMALADE
第37部　テキアペ作成ツールCONTEX
■87年3月号
第38部　魔法使いはアニメがお好き
第39部　アニメーションツールMAGE
付録　“SWORD” 再掲載とMAGICの標準化
■87年4月号
第40部　INVADER GAME
第41部　TANGERINE
■87年5月号
第42部　S-OS “SWORD” 変身セット
第43部　MZ-700用 “SWORD” をQD対応に
■87年6月号
インタラプト　コンパイラ物語
第44部　FuzzyBASICコンパイラ
第45部　エディタアセンブラZEDA-3
■87年7月号
第46部　STORY MASTER
■87年8月号
第47部　パズルゲーム碁石拾い
第48部　漢字出力パッケージJACKWRITE
特別付録　FM-7/77版S-OS “SWORD”
■87年9月号
第49部　リロケータブル逆アセンブラInside-R
特別付録　PC-8001/8801版S-OS “SWORD”
■87年10月号
第50部　tiny CORE WARS
第51部　FuzzyBASICコンパイラの拡張
第52部　X1turbo版S-OS “SWORD”
■87年11月号
序論　神話のなかのマイクロコンピュータ
付録　S-OSの仲間たち
第53部　もうひとつのFuzzyBASIC入門
第54部　ファイルアロケータ＆ローダ
インタラプト　S-OSこちら集中治療室
第55部　BACK GAMMON
■87年12月号
第56部　タートルグラフィックパッケージTURTLE
第57部　X1turbo版 “SWORD” アフターケア
　　ラインプリントルーチン
特別付録　PASOPIA7版S-OS “SWORD”

1988

■88年1月号
第58部　FuzzyBASICコンパイラ・奥村版
付録　石上版コンパイラ拡張部の修正
■88年2月号
第59部　シューティングゲームELFES
■88年3月号
第60部　構造型コンパイラ言語SLANG
■88年4月号
第61部　デバッギングツールTRADE
第62部　シミュレーションウォーゲームWALRUS
■88年5月号
第63部　シューティングゲームELFES II
第64部　地底最大の作戦
■88年6月号
第65部　構造化言語SLANG入門(1)
第66部　Lisp-85用NAMPAシミュレーション
■88年7月号
第67部　マルチウィンドウドライバMW-1
連載　構造化言語SLANG入門(2)
■88年8月号
第68部　マルチウィンドウエディタWINER
■88年9月号
第69部　超小型エディタTED-750
第70部　アフターケアWINERの拡張
■88年10月号
第71部　SLANG用ファイル入出力ライブラリ
第72部　シューティングゲームMANKAI
■88年11月号
第73部　シューティングゲームELFES IV
■88年12月号
第74部　ソースジェネレータSOURCERY

1989

■89年1月号
第75部　パズルゲームLAST ONE
第76部　ブロックゲームFLICK
■89年2月号
第77部　高速エディタアセンブラREDA
特別付録　X1版S-OS “SWORD” <再掲載>
■89年3月号
第78部　Z80用浮動小数点演算パッケージSOR
　　OBAN
■89年4月号
第79部　SLANG用実数演算ライブラリ
■89年5月号
第80部　ソースジェネレータRING
■89年6月号
第81部　超小型コンパイラTTC
■89年7月号
第82部　TTC用パズルゲームTICBAN
■89年8月号
第83部　CP/M用ファイルコンバータ
■89年9月号
第84部　生物進化シミュレーションBUGS
■89年10月号
第85部　小型インタプリタ言語TTI
■89年11月号
第86部　TTI用パズルゲームPUSH BON!
■89年12月号
第87部　SLANG用リダイレクションライブラリDIO.LIB

1990

■90年1月号
第88部　SLANG用ゲームWORM KUN
特別付録　再掲載SLANGコンパイラ
■90年2月号
第89部　超小型コンパイラTTC++
■90年3月号
第90部　超多機能アセンブラOHM-Z80
■90年4月号
第91部　ファジィコンピュータシミュレーションI-MY
■90年5月号
第92部　インタプリタ言語STACK
■90年6月号
第93部　リロケータブルフォーマットの取り決め
第94部　STACK用ゲームSQUASH!
第95部　X68000対応S-OS “SWORD”
特別付録　PC-286対応S-OS “SWORD”
■90年7月号
第96部　リロケータブルアセンブラWZD
■90年8月号
第97部　リンカWLK
■90年9月号
第98部　BILLIARDS
■90年10月号
第99部　ライブラリアンWLB
■90年11月号
第100部　タブコード対応エディタEDC-T
■90年12月号
第101部　STACKコンパイラ

1991

■91年1月号
第102部　ブロックアクションゲームCOLUMNS
■91年2月号
第103部　ダイスゲームKISMET
■91年3月号
第104部　アクションゲームMUD BALLIN'
■91年4月号
第105部　SLANG用カードゲームDOBON
■91年5月号
第106部　実数型コンパイラ言語REAL
■91年6月号
第107部　Small-C処理系の移植
■91年7月号
第108部　REALソースリスト編
■91年8月号
第109部　Small-Cライブラリの移植
■91年9月号
第110部　SLANG用NEWファイル出力ライブラリ
■91年10月号
第111部　Small-C活用講座（初級編）
■91年11月号
第112部　Small-C活用講座（応用編）
第113部　MORTAL
■91年12月号
第114部　Small-C SLANGコンパチ関数

1992

■92年1月号
第115部　LINER
■92年2月号
第116部　シミュレーションゲームPOLANYI
■92年3月号
第117部　カードゲームKLONDIKE
■92年4月号
第118部　オプティマイザO80実践Small-C講座(1)
■92年5月号
第119部　COMMAND.OBJ実践Small-C講座(2)
■92年6月号
第120部　COMMAND.OBJ2実践Small-C講座(3)
■92年7月号
第121部　関数リファレンス実践Small-C講座(4)
■92年8月号
第122部　ワイルドカード実践Small-C講座(5)
第123部　グラフィックライブラリ GRAPH.LIB
■92年9月号
第124部　O-EDIT&MODCNV
■92年10月号
第125部　SLENDER HUL実践Small-C講座(6)
■92年11月号
第126部　EDIT実践Small-C講座(7)
■92年12月号
第127部　MAKE実践Small-C講座(8)

1993

■93年1月号
第128部　EDC-Tの拡張
■93年2月号
第129部　BLACK JACK
■93年3月号
第130部　シューティングゲームコアシステム作成法(1)
■93年4月号
第131部　シューティングゲームコアシステム作成法(2)
■93年5月号
第132部　シューティングゲームコアシステム作成法(3)
■93年6月号
第133部　REVERSI
■93年7月号
特別付録　MSX用S-OS “SWORD”
■93年8月号
第134部　MACINTO-C再掲載
■93年9月号
第135部　7並べ
特別付録　SLANG再々掲載
■93年10月号
第136部　シューティングゲームコアシステム作成法(4)
■93年11月号
第137部　S-OSで学ぶZ80マシン語講座(1)
■93年12月号
第138部　エディタアセンブラREDA再掲載

第6回アマチュアCGAコンテスト

プロジェクトチームDōGA
かまた ゆたか

アマチュアCGAコンテストも今年で第6回となりました。入賞作品は20ページのOh!X Graphic Galleryにて発表しましたが，ここではコンテスト全体の講評とビデオの申し込みについての説明を行います。

はじめに

「今月は，連載休みとちゃうんか」とか，「謎の計画の件は結局どうなったんや」とかいわれそうですが，都合の悪い話は置いといて，CGAコンテストの入賞作が決まりましたので，とりあえず報告させていただきます。

CGAコンテストは，単に作品を集めて優劣を決めるというよりは，アマチュアCGA界の1年間の活動成果発表会的な色彩が強いと思います。ですから，ここでは応募作品の分析を行ってみましょう。

応募総数は92作品（昨年比1.5倍）もあり，着実にアマチュアCGAが定着していると感じられ，まずは喜ばしい限りです。応募総数が増えたということは，新人が増えたということですが，審査が終わってみると，上位入賞には新人はたった1名しかおらず，先輩陣もそうやすやすと席を譲るつもりはないようです。とはいっても，先輩陣のなかには昨年の新人という人も多く，ちゃんと新しい世代が育成されているともいえます。

使用機種別に見ると，やはりX68000（CGAシステム）が圧倒的に多く，AMIGAの躍進も目につきます。意外と低迷しているのがMacintoshやPC-9801で，IBM互換機，FM TOWNSは残念ながら1作品もありませんでした。

グランプリは該当作品なし

今回のグランプリは，残念ながら「該当作品なし」という結果になりました。これは，CGAコンテストの予算上の問題……ではありません（冗談になってないなぁ）。

総合的なレベルからいえば，今回のコンテストは「過去最高のクオリティ」と審査員から賞賛の声もあり，かなりの作品が集まっています。ご覧になればわかりますが，入賞，佳作の7作品（特に入賞4作品）は，ほとんどどれもグランプリをとってもおかしくないようなレベ

入賞作品一覧

作品	作者
グランプリ	
該当作品なし	
作品賞	
Switch On Concent Robo	坊野 博典（KMC）
アニメーション賞	
A.B.C.Day	布山 毅
技術賞	
HOUND '93	下岡 正道
エンターテイメント賞	
冥皇龍ベルギウス	腰原 仁志
佳作	
青年と空缶	宍戸 光太郎（あに作人）
COMPOSITION 002	鈴木 陽二郎
CHUN･CHUN･WORLD II	佐野 元
入選	
かえる	長木 功（テーマ研究CG）
恐竜都市－1997のCM	清家 征雄
展覧会オープニングビデオ	由水 桂
KODOKU	中村 雄一
The J-POP	宗戸 一眞
少女キャレット	加藤 雅敏
XEBEC	山崎 勇（JELL's TAIL）
Gunner 05	小島 禎樹（Studio Dream Field）
HELL DRIVE	白波瀬 登（白波瀬）
エンゲージ ランデブー	土田 康司
WING CROSS	宮崎 智記
VARIABLE ATTACKER	筑摩 昌則（大阪工業大学GR）
Fractal Atmosphere	布山 毅
THE STORY OF SOAP	立岩 潤三
COLOR	立石 武史
妖精BOONの大冒険	太洞 信也
参考入選	
BLUE RUNNER	土田 康司
サンダーホーク	文月 涼
LOGiN特別賞	
恐竜都市－1997のCM	清家 征雄

ルに達しています。ただ，ずば抜けた作品がなく，グランプリの該当作がなかったわけです。

そのほかの各賞の入賞作品は表のとおりです。また各作品については，20ページからのOh!X Graphic Galleryをご覧ください。

そのほか

コンテストのビデオには，オマケとして，『エンゲージランデブー』作者の土田康司さんの『芸術祭オープニング：BLUE RUNNER』と，『TORNADO』の文月涼さんの新作『サンダーホーク』が収録されています。

また，昨年好評だった「1カット部門」「4カット部門」は，選外になった作品の一部も合わせて，「INTRODUCTION」というコーナーで紹介します。さらに，オープニングCGAは，宇宙人森山さんが制作した『ダイダロス！』。というように，ビデオには最初から最後まで120分ぎっしり詰まっています。

ビデオの申し込み方法

ということで，今年もビデオの配布を行います。今年は，このページについている郵便振替用紙で申し込んでください。

＊　＊　＊

第6回アマチュアCGAコンテストビデオ

形態：VHSビデオテープ（約120分）
金額：1本につき3,000円（実費2,500円＋カンパ500円）
締切：1994年4月30日
発送：4月～6月の予定

注意事項

1)　申し込み方法は，郵便振替のみとします。

2)　表の「払込人住所氏名」の欄に，ビデオの送付先（つまり自分）の住所，氏名，郵便番号，電話番号，それとビデオの申し込み本数を，はっきりとていねいに記入してください。住所などがわかるのはここだけですので，記入もれがないようにお願いします。また，ビデオの本数が記入されていない場合，たとえば2本分の金額でも1本分＋カンパとみなされてしまうかもしれません。

3)　今回は，裏面に「アンケート回答欄」を設けました。アンケートの趣旨や内容については囲みをお読みください。ご協力お願いいたします。また，「通信欄」には，DōGAの活動やコンテストなどに関するご意見，ご要望など，ご自由にお書きください。

4)　作業の円滑化のため，この郵便振替用紙は第6回アマチュアCGAコンテストビデオの申し込み専用とします。過去のコンテストのビデオや，CGAシステムのマニュアル，CGAマガジンなどを申し込むことはできません。

5)　ビデオの代金は1本につき3,000円（実費2,500円＋カンパ500円）です。また，それ以外のカンパも同時に受け付けますので，1)にもある通り，申し込み本数は忘れ

500円の攻防

2月○日。

連日，与野党の激しい攻防が繰り広げられる国会議事堂。その隣で行われるCGAコンテスト。実は，コンテスト事務局のなかでも与野党の攻防が行われていた。

連立与党＝CGAコンテスト実行部（実行委員長：まりお，ビデオ総合プロデューサー：おとん，大蔵省：つね，その他）と野党＝かまたでである。本日の議題は，コンテストのビデオの配布価格についてである。

＊　＊　＊

連立与党：今回の配布価格は1本3,000円としたい。

野党：異議あり！　昨年500円の値上げをしたばかりだというのに，今年さらに500円の値上げをするというのは，国民（読者）の理解が得られない。昨年と同様2,500円とすべきだ。

与：昨年と今年では，状況が違いすぎる。

まず，ビデオは昨年90分だったのが，今年は約120分になることが予想される。

野：ビデオテープ代は，90分も120分もそんなに変わらないはずだ。

与：確かにそうだが，ダビング料金も違う。また，マスター制作にかかるスタジオ使用料も当然増額される。さらに，根本的な問題として，90分ならマスターをベータカムで制作できるが，120分用のベータカムのテープが存在しないため，今回からはD2を使用することになる。するとたとえばD2の場合だと，テープ1本で5万円もする。

野：その件については，ビデオ制作の総額見積を提示せよ。

与：（見積を制作しながら）ビデオの制作費以外に，今年から郵便料金が大幅にアップされた。また，大蔵省より次のような答申もある。

大蔵省：はい，昨年現金書留による申し込みを行ったところ，大きなトラブルが発生し，多くの方々にご迷惑をおかけしました。そこで，今年は郵便振替にし，また，宛名書き作業は外部の業者にまかせることにしたいと思います。

与：以上のように，今年から宛名書きなどの発送手数料が加わる。

大蔵省：もっとも，全体としては国民の負担はむしろ減ります。なぜなら，現金書留に比べ，郵便振替は手数料が非常に安いからです。したがって，たとえ500円の値上げとしても，国民の負担そのものは，合計で150円程度しか変わりません。

（ここでビデオ制作の見積ができあがる。一時中断）

野：昨年と同数程度の申し込みがあったとして，ビデオ1本あたりの実費を計算してみた。ビデオ制作費，ダビング費，発送手数料，郵送費などで，2,500円前後になる。昨年は，実費2,000円＋カンパ500円で2,500円としていたが，今年はカンパをなしにして，実費のみとすれば，値上げをしなくてすむはずだ。

与：その計算はおかしい。コンテスト自体，入場料もなく，ビデオの販売以外にその経費をまかなえない以上，コンテストの開催費も，ビデオの実費として計算に入れるべきだ。

野：それは従来の政府の見解と大きく異なる発言だ。実費という以上，ビデオの制作，配布にかかる費用だけのはずだ。白紙撤回せよ。

（会場が荒れだす）

与：ならば，野党は，コンテストを開催するための具体的な財源を挙げよ。

ずに記入してください。

6) 申し込み期限は，1994年４月30日です。お早めにお願いします。

7) ビデオの発送は４月～６月の予定ですが，申し込み多数の場合，または不慮の事態によって遅れることがあります。その場合はOh!X誌上にて告知いたします。

＊　＊　＊

以上の注意事項をよくご確認のうえ，お申し込みください。正しく申し込まれていない場合は対応いたしかねますのでご注意ください。

アンケート

本文でも説明しましたが，ビデオの申し込みの際に，以下のアンケートへのご協力をお願いいたします。アンケート回答欄は郵便振替用紙の裏面にあります。

［問１］

「500円の攻防」を読み，ビデオの販売価格について次のなかから最も適当だと思うものを選んでください。

a）2,500円：ビデオの実費以外のコンテスト運営費のカンパを募るのは不当だ
b）3,000円：コンテスト運営費の一部を補填する程度のカンパはよい
c）3,500円：コンテストの開催全体の採算がとれ，健全な運営を行うべきだ

［問２］

ついにCGAシステムのマニュアルがなくなってしまいましたが，それについては，どうお考えですか。

a）すでに持っているので別に構わない
b）まだ入手していないので，いまのものを増刷してほしい
c）すでに持っているが，新バージョンのマニュアルを作るなら，また申し込む
d）バージョンアップが相当大幅でない限り，申し込まない

［問３］

汎用性のあるデータベースを構築しようとしていますが，どのような内容のものがよいですか。（複数回答可）

a）大理石，惑星，樹木などのマッピング用画像データ集
b）いろいろな風景の背景用画像データ集
c）家やビルなど建築物の形状データ集
d）老若男女，いろんな表情の顔のデータ集
e）車，飛行機などのメカのデータ集
f）家具などの模様替えシミュレーション集
g）ロボットの形状データ集
h）CGA作品につけるBGMデータ集

［問４］

プロジェクトチームDōGAが賛助会員を募集するとしたら，応募しますか。なお，賛助会員とは，会費もなし，義務もないけど決議権もなし，脱会は自由。ただ「DōGAの活動に賛同し，協力の意志がある方」という条件がつきます。

a）絶対に参加する
b）きっと参加する
c）参加するかもしれない
d）参加しないと思う
e）これだけでは，わからん

＊　＊　＊

なお，通信欄には，DōGAの活動やコンテストなどに関するご意見，ご要望などなんでもご自由にお書きください。

野：年間の活動費を切り詰め，コンテストの運営費も切り詰めることで……。
与：もともと切り詰めている運営費のどこを削減できるんだ！　人件費はゼロなんだぞ。
野：しかし，従来の見解と異なる。
与：だいたい，スタッフを大阪から東京会場まで「青春18キップ」（値段が安いが，各駅停車しか乗れない）で行かせるなんてひどすぎるぞ！

（会場が荒れて，審議は一時中断。与野党トップ会談が開かれる）

野：トップ会談の結果，コンテストの開催費もビデオの実費に含めるという発言は白紙撤回する。しかし，コンテスト開催費の一部をビデオ配布の際，カンパという形で集めるのは認めることとする。
与：カンパの金額ですが，コンテスト開催費も昨年より経費がかさんでいます。
野：安易な経費の増額は認められない。経費削減に努力せよ。
与：当然，努力は行っております。多少古くても安い会場を探したり，東京まで送るスタッフを減らすため，現地スタッフを登用したり。しかし，会場は年々大きいところが必要となりますし，現地スタッフに任せられる仕事は限られています。
野：では，どのような点で開催費が上がったのか？
与：まず，会場が広くなって，従来シャープから借りていた液晶プロジェクターでは暗すぎるため，今回から業務用の本格的なプロジェクターをレンタルします。また，例年スタッフが利用していた安宿が今回は使えなかったため，宿泊費が増大しています。さらに，今回から大阪会場の上映会を，ちゃんとホールを借りて行いますので，その会場費などが大きく響きます。
野：今回はグランプリがなかったため，賞金は減ったのではないか。
大蔵省：計算してみればわかりますが，グランプリがなくても，入賞者数は増えていますので，ほとんど変わりません。
与：経費が増大している以上，カンパも増大せざるを得ない。よって，カンパを1,000円としてはどうだろうか。
野：異議あり。最初に3,000円という金額が提示されていたのに，いまの段階で3,500円となるのは，納得がいかない。その数字には根拠がない。
与：しかし，実費2,500円＋カンパ500円では，CGAコンテスト全体は赤字になってしまう。
野：その発想は，コンテストの開催費のすべてをビデオの販売でまかなおうというものだ。もともと，コンテストの開催費とビデオの配布は別ものだったはずだ。
与：だが，完全に赤字のコンテストは，いずれ運営できなくなる。コンテスト開催全体で収支を合わせるのが，健全な運営だ。
野：それは，コンテストだけ独立して採算をとるのではなく，DōGAの活動全体として採算を合わせればよい。活発に活動し，広くカンパをいただき，それを集めてコンテストの資金とすべきだ。
与：しかし，活動成果やカンパは年によって変動の波が大きく，安定した財源とはいえない。
野：いや，断固として，その考えには反対する。あくまでコンテスト開催費の一部を補填する程度のカンパに抑えるべきだ。
与：いや，国民もそのへんの事情をちゃんと解説すれば，納得するはずだ。
野：それなら，ここは，国会を解散し，総選挙に持ち込み，国民の意見を問うべきだ。
与：おうっ，解散だ！　総選挙だ！

＊　＊　＊

ということで，ビデオの販売価格は，一応今年は実費2,500円＋カンパ500円としますが，今後の方針についてはみなさんのご意見を聞きたいと思います。また，ついでにいろいろとご意見をお伺いしようと，アンケートを用意しました。アンケート回答欄はビデオ申し込み用の郵便振替用紙の裏面にあります。みなさん，ぜひ記入してください。

Wave Blaster接続計画

Taki Yasushi 瀧　康史

いきなり始まった怪しい新連載。環境改善のための比較的簡単な回路を扱います。たま氏が「これつながるんじゃない?」といって，編集者が確認もせず買ってみて，「瀧君，がんばってね」で始まったこの計画。さて……?

ついこのあいだ，巨大な仕事をやっと終えたかなぁなんて思ったら，いつのまにやら，なにかやることになってしまいました。

今回から，毎月やるのかときどきやるのか謎ですけど，とりあえず役に立つローテク（ときにハイテク）ハードウェアの連載を始めたいと思います。

コンセプトは，「役に立つ」ということ。したがって，あまりハードウェアのお勉強とか，そういう意味あいを込めるつもりはありません。また，できる限り，連載1回でひとつのハードを作り上げるつもりです。今後，でっかいのを手がけないって保証はないですけれど。1回ぽっきりで作れたほうがこれ便利そうだな，じゃぁ作ってみようといったノリでいけるんじゃないかと思ったからです。

必要最低の知識は回路を読めることです。

しかし，回路にもいろいろあって，大雑把に分けると，中学校の技術でやるぐらいの回路しか読めない人，ICを使った簡単な回路がとりあえず読める人，論理回路がわかる人，回路図の暗黙の省略がわかる人などといった具合に，ひと口に回路といっても対象によって異なってしまいます。

では，中学レベルの回路で通せばみんなに伝わるといえそうですが，困ったことにこれは大は小を兼ねるわけではなく，中学レベルの回路図では，ひと目でなにをしている回路であるかわかりません。

たとえば論理回路などで，いちいち74HC004の何番ピンと何番ピンをつなぐと書かれていたとします。これだと初心者にはつなげ方がわかるでしょう。しかし，回路図を「見て」作成する人ではなく，回路図を「読んで」作成する人には，苦痛でしかありません。

そこで，回路図は一般に出回っている，暗黙の省略は省略として扱うレベルのものを使います（今回，オペアンプに電源が書いてありませんが，各自電源はもちろんつなげること）。理由は，私自身にも，自分の作った回路に自信がないからです。わかりづらいんじゃないか？と思われる部分はある程度まで本文中に注を入れます。

また，基本的には，動けばいいや的な発想で回路を書きます。もちろん，細心の注意を払って，間違いが起きないように，書くつもりです。ですが，もし誤りを発見した場合，できる限り早めに間違いを教えてほしいところです。

そして，今後掲載されるすべての回路の著作権を放棄する代わりに，責任も放棄させていただきます。つまり，掲載された回路を作成することで，あなたのコンピュータが壊れてしまったなどの責任は，すべて各自で取ってください。これに関して，編集部も私も一切責任が取れません。

それから，もし，こんなことができたらいいな？みたいな発想がありましたら，どしどしおはがきください（もっとも早いサポートは，PC-VAN X1CLUBのフォーラム内ハードウェアで行っております。SIGOPが私なのできてくれるとありがたいんですけどね……)。

平均所持音源向上計画その1

皆さんはX68000に特別な音源をつけていますか？　外部音源といえば，MIDIからSC-55をつないでいる人が大半でしょう。

本誌では当たり前のようにMIDIを扱っていますが，実際のところ，まだまだMIDI楽器というものは一般的ではありません。聞くだけの人にとってMIDI音源はまだ高価なものですし，パソコン通信などをしていない一般ユーザーにとってはゲームとOh!X LIVEに掲載される少しばかりのデータを聞くためのアイテムでしかないのですから。そのために楽器とインタフェイスをあわせて6万円弱なんてなかなか財布は割れないですよねぇ？

そこで，もっと安く，そこそこによい音源が手に入らないものかといろいろ考えていたところ，楽器方面ではなく，意外にもAT互換機の音源ボードのほうによいものが転がっていました（協力：たまたまき氏)。

実際のところ，X68000の音源はまだまだまともなほうです。PC-9801方面はOPNAな86ボードなるものが出てきて，平均的な性能はいまのX68000の内蔵音源よりもまっとうになってはきています。しかし，実際ゲームなどから奏でられる音楽は，たとえ86ボード対応と謳っていても，文化の違いか，どれもチャカポコとした，数年遅れのゲームミュージックなものがたくさんあります。つまりどれだけX68000の内蔵音源の駆使のされ方が素晴らしいか！

でもX68000の音源に不満を持っている人はたくさんいますよね？

そこで，もともとAT互換機のものですが，PC-9801にもある，Sound Blasterという音源ボードの上にジョイントされるWave Blaster（以後WB）をなんとかX68000につなげないかと考えてみました。

この音源のコアとなるICとPCMデータはE-muのものです。わかる人は「ぴぴぴ」ってきますよね？　規格的には実は本物のMIDI楽器で，GM規格準拠。ボイス数は

Wave Blaster
東海理化販売　☎03(3580)3003

32。砕いていえばSC-55相当なんですが，エフェクトはありません。でも代わりにボイス数が多いですけどね。SC-55内蔵のような玩具のようなエフェクトがついてるよりも，あとで自分でエフェクタを買ってきたほうが音がよいでしょ（チャンネルごとに設定はできませんが）。でも，パーシャルはあとから増設できません。なのにお値段は23,000円程度。安い店で買えば，2万円強ぐらいで買えちゃいます（編集部のものはT・ZONEで20,800円だった）。やっぱり市場が広いのは強みですなぁ。

SC-55相当の音源が2万円だったらどうします？ MIDIボードと買っても3万ちょっと。もっとケチれば，Z-MUSICを使う限りRS-MIDIで済ませられるので，音源代だけしかかかりません。SC-55の曲をそのまま聞くにはちょっと難点もありますが，その他一般のGM楽器に比べれば十分，なんとかなる範囲ではあります。

平均所持音源向上計画。成功するといいですなぁ……。

部品説明

とりあえず音源そのもののレビューはあと回しにして，ものを作ってみましょう。

WB側のピンアサインは資料でわかったものの，相手側の定格がわからないので，見当進めで基板を作っていきました。写真ではむちゃくちゃな配置になっていますが，これでもちゃんと音は鳴ります。

写真ではオペアンプの回路が別基板になっていますが，皆さんは一緒にしたほうがよいでしょう。裏面のハンダ面は秘密。配線をとったりつけたりしたので，ものすごく汚いですから。

とりあえず右下の表を見てください。

これが部品表です。お金がかかりそうなのは電源ぐらいなもので，電源をどっかから調達できれば，全部で1000円程度でできてしまうかもしれません。

順に説明していきましょう。

1はWB。これは当たり前に必要です。

2は電源。+5V，±12Vのものが必要です。ただ，もともとアドオンボードなので，それほど電流をとっているとは思えません。±12Vはオペアンプの電源にも使うので，それなりに安定しているほうが音がよいでしょう。ちなみに編集部では，+5Vは3A。±12Vは2A前後のものを使用しています。

本来，これらを測定して掲載するのが親切心なんでしょうが，ちょっと時間不足で，定格の測定まで手が回りませんでした。

ごめんね。

なお，今後ゲーム基板を購入する予定がある人は，+5Vは7Aから8Aぐらいのものを選ぶのも手です。もっとも，音源専用に電源があったほうが音がよいに決まってますが。

また，自分ですでに自作ボードを作ってしまっている「つわもの」は，本体の拡張スロットから電源を盗み出してくるのもよいでしょう。そういう人に対してはいらぬお世話でしょうが，コンデンサつけて±12Vのノイズはできるだけ除去してね……といっておきましょう。

これがWave Blaster本体

3はWBとこれから作るWBのいわばインタフェイス基板とをつなぐコネクタです。WBを先に買った人にはわかるでしょうが，コネクタというよりもジャンパスイッチなどで利用されているような代物です。13×2列のものを購入してくればよいでしょう。

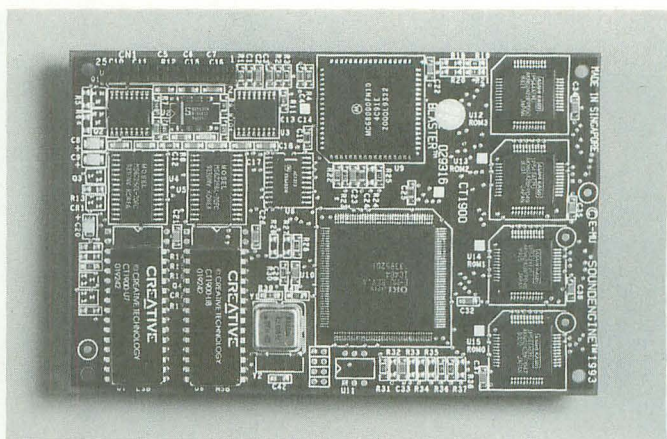
なお，これらはニッパーでブッチ切ることができるので，2列のものであれば，長いに越したことはありません。のちのちジャンパスイッチに使えますし。

4は抵抗です。アナログ回路上にあるわけではないので，ローノイズの高級品を買っても，音がよくなるとかそういうことはありません。

5はコンデンサです。回路上で指定されてるのは，47μF/16Vだけで，あとはパスコンです。ちなみに，写真の基板では100μF/10Vがついてたりします（単に自分の家に在庫がなかっただけ）。コンデンサなんてデカけりゃいいという考えは真似しないように(大きすぎると，壊れるのが早い)。5Vしか電圧はかけていませんが，実は16Vぐらい耐圧はあったほうが安心です。

6はダイオードです。1S1588というわりとポピュラーなダイオードですから，200個ぐらいまとめて買っておきましょう。こういってるってことは，あとでよいことがあるということですよ。うん。

7はフォトカプラです。半導体関係を扱っている店に売っています。地方の方には入手が難しいかもしれませんが，絶対なきゃダメです。

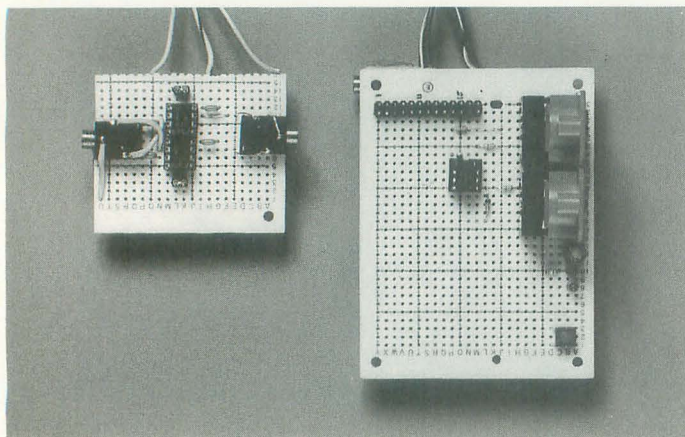
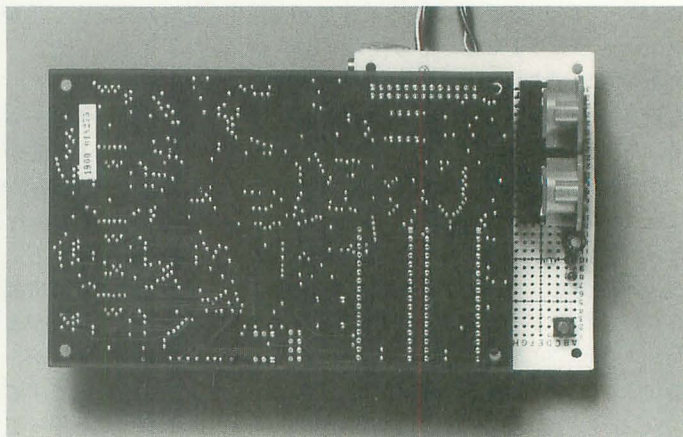
8は5ピンのDINです。MIDI端子用ですから，間違えることはないでしょう。

MIDI THROUGHがいらなければ，ひとつでもかまいません。でも抵抗1個とコネクタひとつ節約しても大した金額にはなりませんね。コネクタはできる限り基板にハンダづけできるタイプのものを買ってきてください。

オマケですが，同じようにサクサク並列につなげていけば，当然MIDI THROUGHがたくさんできます。まぁMIDI楽器を複数持っている人はたいていMIDIパッチベイあたりも持っているでしょうが（ちなみに，現在，編集部にも私の家にもありません）。

9はAUX出力用コネクタです。写真ではミニフォンジャックなるものを使っていますが，AUXのピンジャックで基板に接続できるものが入手できるならば，そっちのほうが便利かもしれません。ちょっと探した限りでは，見つからなかったので，私は諦めてこっちにしました。

なお，写真では謎の白い物体でミニフォンジャックのメスがくっついていますが，これはホットボンドというものです。鉛筆状の長いホットボンドの素を，機械の後ろから挿入すると，先からとろとろと出てき

部品表

1．Wave Blaster本体
2．電源+5V，±12V
3．ジャンパスイッチを接続できる13×2列のコネクタ
4．抵抗
　270Ω　1個
　220Ω　2個
5．コンデンサ
　47μF/16V（電解）　1個
　パスコン用0.01μF　5個
6．ダイオード
　1S1588　1個
7．フォトカプラ
　PC900　1個
8．5ピンDIN（MIDI端子用2個）
9．AUX出力用コネクタ
10．オペアンプ　LF356　2個
11．リセットボタン　1個

完成した2つの基板

Wave Blasterをつないだところ

て，しばらくすると固まる……といったものですが，わりとなんでもくっついちゃうし，部品の固定には便利なので，お金に余裕のある方は買ってみるとよいかもしれません。

10はオペアンプです。ステレオなのでLF356を2つ使用しています。回路上では省略していますが，±12Vの電源を必ずつけてください。それぞれに2つずつ，パスコンをつなげるのがよいでしょう。

写真上でLF356の隣にひとつショートしている2Pソケットがありますが，もともと増幅回路にしようか，ボルテージホロアにしようか迷ったからそうしているだけで，作る人はいきなりつないでください。

LF356を利用した理由は，入力にバイポーラFETを利用しているので，入力バイアス電流が低いとか，内部位相補償型でありながらスルーレートは12V/μSもあるとか，もっともらしいことはいくらでも書けるのですが，実は単に，有名でよく使われているオペアンプなので，多量に買い込んでいただけという説が有力です。

どうせただのボルテージホロアなので，自分がいつも利用するオペアンプがある人はそれを使っても一向にかまいません。

まず，回路左側はWBのピンアサインです。WB側に番号がプリントされているので，それをちゃんと見て配線してください。

11は単なるリセットスイッチです。ですからプッシュスイッチならなんでもかまいません。なくたって動きますが，音源が鳴っているときに本体のリセットかけると，ときどき鳴りっぱなしになっちゃうので，あったほうがよいでしょう。ただのプッシュスイッチなら20円ぐらいで売ってるんですから。

部品表には書いてありませんが電源をON/OFFをするスイッチもあったほうがよいと思われます。なぜなら，現在使っていて非常に不便だからです。写真ではそんなものというか，電源すら写っていませんが，私は原稿を書き終えたあとに，スイッチを作ろうと心に決めました。

それから，それぞれ違う色のLEDを2つと500Ωぐらいの抵抗をひとつ買っておくと，楽しみが増えます。別に好きな色で結構ですが，ひとつは高輝度タイプが理想的です(青には高輝度タイプはないよん)。とはいえ，赤はずっと見ているとまぶしいので，赤の普通のと緑の高輝度タイプがよいでしょう。

回路上には記載していませんが，MIDI INの4,5ピンのあいだにLEDを入れるとMIDI-SENCEがきたときに，CM-64のようにチカチカ光ります。この接続に根拠はまったくなく，そうやって接続したら，LEDがちゃんと光りましたし，データが破壊されてるようでもなかったのでそのままにしてあるだけです。多分，問題ないでしょう。きっと。

回路説明

図1が実際の回路図です。

左のコネクタがWB。右がMIDI IN，MIDI THROUGH，AUX出力です。

WBの1から25の奇数(つまり片側)は全部GNDなんでひっつけても構いません。錫メッキ線かなんかでピタっと接続しておきましょう。

2番はリセットボタンです。GNDに落とすだけなんで，たいしたことはありません。

4，8はラインアウトです。右左をそれぞれLF356まで引っ張ればよいのですが，これはできるだけ短いほうがノイズがのりません。なんていいながら写真では思いきり延ばしてますが，製作時にアナログ部分とデジタル部分は分けたほうが，回路の変更が楽なのでそうしただけです。皆さんは密着して接続しましょう。

恐しいことに隣のピンは±12Vですから注意が必要です。5Vでは大したことないのですが，±12Vになると確実にICを壊します。間違えないように，この部分には細心の注意を払ってください。私は次回用の回路を作りつつ，読者の皆さんの大切なオーディオ機器や買ったばっかりのWBが壊れないように祈っております。

回路上でWB側に±12Vが接続されてないように見えますが，これは，ちゃんと接続してくださいね。LF356にも間違えずに±12Vの電源を供給するように。

14，18，22はVcc (+5V) です。全部つなげるに越したことはないのですが，ひとつだけでもWB側で3つは接続されているので，ちゃんと動作します。

24番はMIDI INです。この信号の処理が多少手間取ります。多少間違えても，壊れはしませんが，間違えないようにしてくださいね。フォトカプラを中心に部品配置を考えてください。

最終的に，2，4，6，8，10，14，18，22，24がそれぞれ，変に接触していないか確かめてください。特に±12Vが流れる6，10ピンとなにかが接触してないか注意するように。

WBを差す前に一度電源を入れてみて，フォトカプラやLF356が変に熱くならないかチェックしてください。電源ピンにちゃんと電源がきているか，そういうことを確実にチェックしてくださいね。

これで完成です。

やっていることを簡単に説明しましょう。「つわもの」や知識はいらない人は次に進んで，蘊蓄として知りたい人だけが以後を読んでください。

フォトカプラは電気的には絶縁して，信号だけは伝えるというものです。もしMIDI INから不当な電圧，電流の信号が流れてき

たとき，音源を保護するためにあります。

LF356は単なるボルテージホロアです。まぁこれもフォトカプラと似たようなものなのですけどね。WBから流れる電流が非常に少ないということによります。

接続するオーディオの入力インピーダンス（抵抗）が少ないと，電流がいっぱい流れてしまいます。オームの法則ですね。出力電圧は一定，入力抵抗は一定だとすると，I=V/RでRが少なければ，Iは異常に増えてしまうのは目にみえてます。

しかし，WBにはそれに応えるだけの電流を流せないので，その結果，電圧が低下します。すると信号そのものが弱まってしまうので，外にいく信号が実質的にないに等しくなってしまうのです。

そこでボルテージホロアの出番です。

ボルテージホロアの入力インピーダンスは理想的に無限大なので，WBからは電流は0しか流れません。実際には電流が流れないと信号がつながらないので，ほんのわずかだけ流れるのですが，それでもこれによって，WBの負担はずっと軽くなります。

ボルテージホロアは電圧信号だけはそのまま出力に出しますが，電流はたくさん流すことができます。したがって，接続されるオーディオ機器，中間にはいるコード（これにも抵抗はある）があっても，信号がうまく伝わるというからくりなのです。

興味がある人は専門書をご覧ください。

実際の音を聞いてみる

それがなかなかいいんですよね。

「これが約２万円の音源かぁ！」って唸ってしまうほど。

いくら安いとはいえ，もとはE-muのROMなんですから，よいに決まってますけどね。

音色により得手，不得手がある楽器なのですが，得手はブラス，ギターだといえるでしょう。SC-55に比べると，ブラスは歯切れがよく，音がはっきりしています。録音バランスがSC-55に比べて若干大きいため，SC-55に合わせた曲は，ブラスが目立ってしまいますが。

また，ギターはギターでナイロン弦は艶やかな音を奏でますし，ディストーションや，オーバードライブも，わりとよい音を出してくれます。ハーモニクスはイマイチでしたけど。

ベースはチョッパーがなかなかよい味を出しています。ドラムもSC-55に比べると，スナップのきいたよい音です。ピアノは，クラシックには向いてませんが，ロックピアノらしい音です。

要するにロック系の音楽は，SC-55で鳴らすよりも，遙かによい音がWBで楽しめます。もちろん，これはWBでバランスを取り直した場合なのでSC-55にあわせてある既存のデータを聞く場合，多少バランスとりが必要になります。

全体的に，さすがアメリカの楽器……という感じです。しかし，アメリカ人は興味がないのかなぁ？と思ってしまうのが，まずオルガンです。オルガンはSC-55のほうがまだましです。それと木管楽器。アクセントのない，ふにゃっとしたフルートやピッコロやクラリネットなどをなんとかしてくれぇって感じ。あんまり気にしないのかなぁ。

その他は，可も不可もなしといったところ。全体的に見て，SC-55は平均的にバランスが取れてる分，すべてに不満が出てしまいますが，WBはロックを作るならばかなり満足のいく楽器になる反面，クラシックにすると，木管でコケてしまうという難点があります。

もっとも，まだWBの可能性をすべて味わったわけではないので，あまり大きなことはいえませんが，あながち筋違いな意見ではないことは確かでしょう。

でも，音の肌理はSC-55より細かい感じがします。WBはエフェクトがない分，原色で勝負してるって感じです。16ビットサンプリングですし，サンプリングレートもひょっとしたらWBのほうが高いかもしれません（未確認情報）。どちらにしても，１から作るのであれば，WBのほうが（ロック系に限っては）データが作りやすいといえるでしょう。

お買い得度は10ですね。

私事ですが

最初にいったとおり，私もまだまだハードに関しては入門レベルです。ハードウェアの道はまだまだ奥が深く，デジタル回路はパズルみたいなもんなのでわりと作っていましたが，アナログ回路は机上では回路が作れるものの，実際作ってみると作るたびに変わる特性に頭を悩まされています。

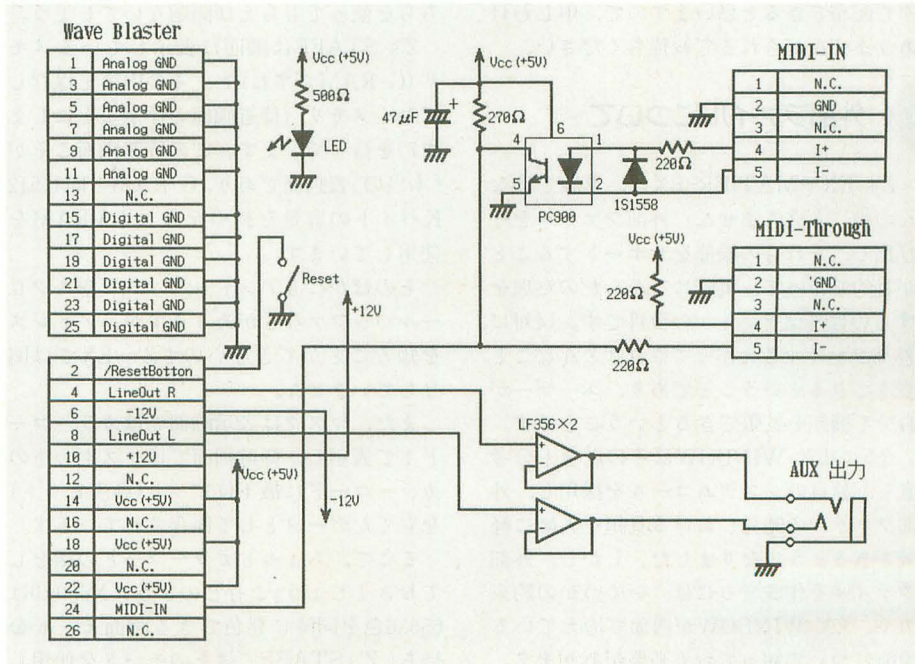
アナログはすべてノウハウといわれていますが，私にはまだまだノウハウが足りないかもしれません。前の音楽の連載とは勝手が違いますが，できる限り，皆さんの役に立つ回路を作りたいと思います。なにか要望がある方は，どしどしはがきをください。

あと，音楽の連載が終わって，１冊の本にしてくれとか，そういうおはがきを，たくさんいただきました。連載中はあまり反応がなかったので，やってる本人は結構不安でしたが，思ってもいない反応にとても喜んでいます。

機会があれば，書き直して１冊の本に仕上げたいのですが……どうでしょうね？

ちょっとその辺は私にはわかりません。

ともあれ，今度はまったくフェイスの違った連載ですが，またしばらくおつきあいください。ではまた。

図１　回路図

Z's-EX & MATIER-EX

外部ファイルの作り方

Kikuchi Isao 菊地 功

EX-WINDOWシステムには対応プログラムを作成するための支援機能が用意されています。ここでは実際のプログラム作成の注意点を見ていきましょう。まずはディスクに収録されていたサンプルプログラムの解説です。

「あれっ？ ここはどこ？ 私は誰？(古っる～)」ってことで，先月は私のせいじゃありません。コンフィグファイルが間違っていたのもディスクにRF.DATとMAPICON.DATが入っていなかったのも文章が支離滅裂だったのも私に憑依した霊がしたことですって馬鹿なこといってると怒られそうなので，ごめんなさい。

ちょっと時間がないといいながら現実逃避してたもんで。今月は大丈夫です，たぶん。とりあえずRF.DAT(FRACTAL用のデータファイル）とMAPICON.DAT(PERSPECTIVEのウィンドウイメージデータ）は以前のZ's-EXのが使えますので，持っている方はそちらを外部ファイルと同じディレクトリに置いて使用してください。

また，持っていない方でもMAPICON.DATに関しては，ウィンドウ表示が壊れますが，適当なファイルをMAPICON.DATという名前にしてやればとりあえずは動くと思います。RF.DATは来月の付録ディスクで配布できると思いますので，申しわけありませんがそれまでお待ちください。

外部ファイルについて

Z's-EXやMATIER-EXは，単独ではなんの役にも立ちません。外部ファイルを呼び出し，それらの機能をサポートすることが目的であって，実際になんらかの処理をするのは外部ファイルの役目です。反対に，外部ファイルさえ作ってやればどんなことでもできるということであり，ユーザーが自分で機能を拡張できるということです。

今回のEX-WINDOWはその点を十分考慮し，独自のシステムコールを採用し，外部ファイルの開発における負担を大幅に軽減できるようになりました。しかし，外部ファイルを作成するには，いくつかの約束事や，EX-WINDOWが内部で抱えている機能について知っておく必要があります。

外部ファイル作成用として，3月号の付録ディスクにC言語のライブラリとアセンブラのマクロを収録してありますが，関数の機能などはドキュメントを見ていただくとして，ここではサンプルを用いてC言語による外部ファイルの開発手順を示していきたいと思います。

内部構造

外部ファイルをサポートするためのルーチンなどは，EX-WINDOWが持っており，外部ファイルは引数とともにTRAP7を実行することにより，それらの処理を呼ぶことができます(EXコールとでも呼ぶことにしましょう)。ちょうど，IOCSコールのようなものだと思ってください。

C言語のライブラリでは，それぞれの関数が自分でTRAP7を発行しますので，特に意識する必要はありません。また，アセンブラでもマクロを用意しましたので，そちらを使ってもらえば問題ないでしょう。

Z's STAFFは画面に表示しているメモリ(G-RAMですね）と，その内容を保存しておくメモリ（待避画面と呼ぶことにします）を持っています。この待避画面こそが「本当の」表画面であり，G-RAMと同じ512Kバイトの容量を持つテキストVRAMを使用しています。

そのほかにもアンドゥバッファやスクロールバッファなどがありますが，アドレスを知ることができないのでZ's-EXでは関与していません。

また，マスクは表示画面ではカラーコード1で表示し，待避画面ではマスクの下のカラーコードに最下位ビット(輝度ビット）を立てたデータとして保存されています。

ここで，ちょっとカラーコードの話をしておきましょう。ご存じのようにX680x0は65536色を同時に発色できる画面モードを持ち，Z's STAFFではそのモードを使用しています。

65536色という，一般人には非常に半端な色数は2の16乗からきてます。コンピュータはデジタルですので，0と1の……という話はもういいでしょう。とにかく16ビットデータで表現されているわけです。これがカラーコードです。

X680x0ではこの16ビットの上位から緑5ビット，赤5ビット，青5ビットそして輝度1ビットの構成になっています。緑・赤・青は問題ありませんよね。一般に光の3原色と呼ばれるものです。では輝度ビットとはなんでしょう？ 早い話がそのビットを立てるとRGBすべての値を0.5上げたことになる，というようなものですが（たぶん)，これはちょっと使いにくいということで，Z's STAFFではマスク情報用に使用しているわけです。ですから，実際にZ's STAFFで発色できるのは32768色ということになります。

話を元に戻しましょう。Z's-EXではさらに裏画面を使用する場合は，待避画面と同じデータ構造でメインメモリから512Kバイト確保し，アナログマスクを使用する場合には表と裏のそれぞれ256Kバイトのマスクバッファを確保します。

また，マスクしてあるかどうかはZ's STAFFと同様に待避画面の輝度ビットで判断しますが，透過率はマスクバッファ内で1ピクセルあたり1バイト（正確には上位5ビット）で表され，画面にはその1バイトデータを上位バイトとし，下位バイトは最下位ビットのみ立っているカラーコードで表現されます。

マスクは0～31で表され，値が小さいほど透過率が低く，0でZ's STAFFと同様のマスクになります。ちょっとややこしいですが，外部ファイル作成時には勝手にマスクを対処してしまう関数も用意していますので，それほど心配する必要はありません。

MATIERでは待避画面というものがあ

りませんので，アンドゥバッファを待避画面の代わりに使うことにします。裏画面は最大4画面持てますが，EXOPEN.X起動時にオプションでそのうちの1画面をMATIER-EXの裏画面として登録します。

マスクに関してはZ's STAFFと大きく異なり，表示画面・待避画面ともにマスクの下のカラーコードを反転したデータで埋められます(MATIERもカラーコードに輝度ビットを使用していないので，マスク部分の輝度ビットは立っていることになります)。このままでは外部ファイル中でマスク操作をするときには場合分けをしなければなりません。そこで，MATIER-EXの起動時にマスクをZ's-EX方式に変換しています。これにより，マスク操作をする外部ファイルでもEX-WINDOWの種類を区別する必要がなくなるわけです。

いい忘れましたが，EX-WINDOWではDOSコールのPRINTとIOCSコールのB_PRINTをトラップして，それぞれ確認ウィンドウと選択ウィンドウが開くようにしてあります。PRINTの戻り値はありませんが，B_PRINTは'OK'の場合は0を，'CANCEL'の場合は1を返します。これは不用意にテキストVRAMにアクセスすることを避けるためであり，EXコールのCONFIRMとSELECTでも同様の働きをします。

外部ファイルの作成

では，実際に外部ファイルを作成してみましょう。3月号の付録ディスクに収録されていた簡単なサンプルで説明していきます。

その前にコンパイル環境についてですが，私はコンパイラはGCC，アセンブラはHAS，リンカはHLK，ライブラリはXC付属のものを使用していますので，それ以外のものを使用している場合はmakefileを書き直す必要があります。ただし，上記以外のものを使った場合，正常にコンパイルできるかどうかは保証できません。ご了承ください。

コンパイルするには，環境変数INCLUDEで設定してあるディレクトリにEXLIB.Hを，LIBで設定してあるディレクトリにEXLIB.Lをコピーし，コンパイル時にほかのライブラリと同様にEXLIB.Lを指定してください。ただし，スーパーバイザモードからユーザーモードへ戻しているものをGCCでコンパイルする場合，GCCのバージョンによっては'-fno-defer-pop'オプションが必要です。

(1)ALTERNATE TO GRAM

まずは手始めとして，簡単なところから。裏画面を表画面にコピーしてみましょう。矩形指定を（コンフィグファイルのフラグを立てて）システムで行わせ，その領域内だけを対象とすることにします。ここでのポイントは，

1) 矩形指定範囲パラメータの拾い方
2) 待避画面と裏画面のアドレスの拾い方
3) マスクの判定
4) リターンコード

の4点です。

では，プログラムを見てください(ATOG.C)。まず，矩形範囲を拾っています。EX-WINDOWから外部ファイルを実行する場合，次のような順番でパラメータが渡されます。

裏画面アドレス
矩形範囲左上X座標
　　　　　Y座標
　　　右下X座標
　　　　　Y座標
パラメータ1
　　　　　2
任意オプション

ないものは省略されますが，マウスでどのような順番で矩形指定しても矩形範囲の座標は上記のような順番になります。

ここで矩形指定してあるかどうかはパラメータの数が5個以上あるかどうかで判定できます。プログラム中では，acは外部ファイル自身も個数に含まれますので，6以上かどうかで判別し，それより小さい場合は初期値として画面全体が対象となるようになっています。

次に待避画面と裏画面のアドレスを，それぞれEXコールのBUFFADR()とGETADR()の戻り値として得ることができますので，unsigned short型のポインタで受け取ってください（裏画面アドレスはパラメータでも受け取れますが，関数を使用したほうが無難でしょう)。

その際，裏画面が使用できない場合は，GETADR()は0を返します。この外部ファイルでは，裏画面を使用できない場合はお話になりませんので，メッセージを確認ウィンドウで表示して終了させています。ここで，終了コードは2になっていますが，これは終了後のEX-WINDOW側でとくになにも処理をさせない場合です。終了コードとシステム側の処理の関係は，

0…G-RAMを待避画面にコピーします。システムで矩形指定をした場合は，その矩形内のみをコピーします。

1…待避画面からG-RAMを復元します。エラーが発生し，G-RAMを元に戻したい場合などに指定します。

2…なにもしません。画像を操作する外部ファイルではない場合や，外部ファイル内で待避画面にも書き込んだ場合に指定します。

−1…メモリ操作は行わず，すぐにEX-WINDOWを抜けてZ's STAFFもしくはMATIERに戻ります。ESCキーが押されたときの処理などに使用します。

上記以外…"エラーが発生しました"という確認ウィンドウを開いたあと，1と同じ処理をします。

メインルーチンでは，矩形範囲内を左上から横方向にスキャンして，裏画面からG-RAMにコピーしているだけです。ただし，このときに最下位ビットが立っているかどうかでマスクを判別していますが，表画面は表示画面より待避画面で判別するほうが安全です。

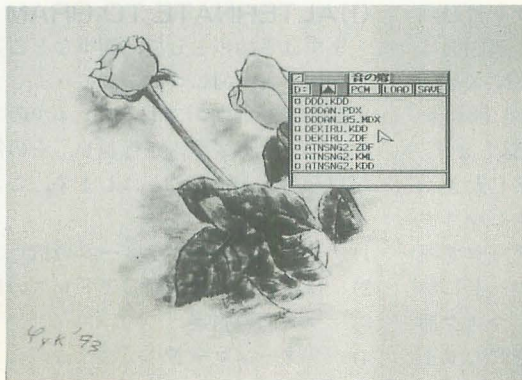
しかし，この外部ファイルでは表か裏どちらかがマスクがしてあればスキップしてしまい，アナログマスクには対応していません。対応させるにはマスクバッファの値から裏と表の演算を行わなければならないのですが，その演算をするEXコールが用意されていますので，そちらを使用してみましょう（ATOG2.C)。

仕様としては，表画面をアナログマスクに対応させる以外は上と同様で，裏はどんなマスクであってもスキップさせることにします。ATOG.Cとほぼ同じですが，G-RAMに対してはいったんマスクを無視して書き込んでしまいます。その後，EXコールCOMGVRAM()を用いてG-RAMと待避画面を合成します。このCOMGVRAM()が実際に行っている処理は，

1) G-RAM，待避画面でともに最下位ビットが0（マスクされていない）のとき，G-RAMの内容をそのまま待避画面にコピーします。

2) G-RAMはマスクされていないが，待避画面はマスクされていたとき，マスクバッファの値とG-RAMと待避画面を演算し，G-RAMにはマスクを，待避画面には演算結果をコピーします。

3) G-RAMがマスクされている場合は，(待避画面がマスクされていようがいまいが）マスクの情報が待避画面とマスクバッファにコピーされます。ただし，マスクされているG-RAMとは，上位バイトがマスクの透過率で下位バイトが01_Hのピクセルのことです。

マスクの反転

ファイラーの例

つまり，この関数を使えばG-RAMに好きなように描いたものがちゃんとよきに計らってくれるわけです。使い方は，EXLIB.Hで定義されているSQUARE構造体に矩形範囲を代入し，COMGVRAM()に渡すだけです。また，この関数は待避画面にも書き込みますので，終了コードは2にしてあります。

これでアナログマスクにも対応できました。いかがですか？　それほど難しくもないということがおわかりいただけたと思います。もっとも，これは基本中の基本ですが。では次に移りましょう。

(2)REVERSE MASK

これまで，Z's-EXはアナログマスクを扱えるようになったとしつこいほどいってきました。ここで，アナログマスクの内部構造についてもう一度詳しく説明しましょう。Z's STAFFのマスクはONかOFFのどちらかでした。それゆえ，マスクの情報は1ビットあれば十分で，待避画面の輝度ビットに保存されていました。

しかし，今回のアナログマスクは5ビット32段階とし，新たにマスクバッファを設けることとしました。マスクバッファは1ピクセルにつき8ビット確保し，実際にはその上位5ビットを使用，0で不可視マスク，大きくなるほど透過率がよくなることはすでに述べました。

しかし，私もあとで「しまった」と思ったのですが，マスクが32段階ではマスクをしない状態を含めて33段階になってしまい，5ビットで表せなくなってしまいます。そこで，マスクをしてあるかどうかは待避画面の輝度ビットで判断し，マスクがしてある場合のみマスクバッファを有効にするという方法をとりました（マスクのない部分ではマスクバッファは0にしておいてください）。

仮に，マスクをしてある部分でマスクの下の色(待避画面の輝度ビットを除いた色)をa，マスクの上から書き込む色をb，マスク透過率（マスクバッファの上位5ビット）をmとすると，待避画面には，

(a×（32−m)＋b×m)/32

という値に輝度ビットを立てた色を入れることになります（実際にはRGB成分に分解/合成する必要があります)。

また，表示画面(G-RAM)には下位バイトが01_H，上位バイトがマスクバッファの色で表示されます。つまり，マスクバッファは上位5ビットしか使用されていませんので，2ビット目から11ビットまでは0で埋められたデータということになります（マスクバッファの下位3ビットに不用意にデータを書き込むと，暴走の原因になります)。

説明はこのくらいにして，実際にマスクを操作してみましょう。サンプルは矩形の内部のマスクを反転させる外部ファイルです(REVMASK.C)。反転の方法は，マスクしてない部分は不可視マスク(透過率0）にし，透過率1のマスクは31にするようにします。また，今回は外部ファイル内部から矩形指定を行ってみました。ここでのポイントは，

1)　マスクバッファアドレスの拾い方
2)　矩形指定の方法
3)　アナログマスクの操作法
4)　アナログマスク使用不可環境の対処

マスクバッファアドレスは待避画面や裏画面同様，EXコールで得ることができ，MASKADR()を使用します。引数は0が表画面，1が裏画面のマスクバッファで，戻り値が0の場合はアナログマスクが使用できないことを示します。

ここで注意しなければならないのは，unsigned char型のポインタで受け取ることです（やったことはありませんが，共用体で上位5ビットを拾ってもいいかもしれません)。

矩形指定をするにはGETAREA()を用います。引数として構造体SQUAERを渡してやると，矩形指定モードとなり，矩形範囲がSQUAERに代入されて戻ってきます。その際の戻り値は0ですが，右クリックによってキャンセルされた場合は−1の値を取ります。プログラムではその場合はループを抜けて終了しています。

さて，メインルーチンでは先ほどと同様に領域の左上から横方向に走査しています。ここで，各ピクセルのマスクを反転させるのですが，前述のようにEX-WINDOWの環境によって処理が異なってきますので，場合分けします。

まず，アナログマスクが使用できない場合(関数digital())，待避画面の輝度ビットを見て，さらにマスクがある場合とない場合に分けます。処理自体はもう説明の必要はないでしょう。コメントどおりです。

次に，アナログマスクが使用できる場合ですが(関数mask8())，マスクをしていない場合は例外処理とします。コメントのとおりですので問題ないでしょう。さらに，不可視マスクの場合も例外的に処理してしまいましょう。

さあ，半透明マスクですが，前述のように反転させるには32から透過率を引いた値を新しい透過率にすればよいことがわかります。プログラムでは透過率に3ビットの下駄を履かせた状態（すなわちマスクバッファの値そのまま）で新しい透過率を計算し，さらにその演算結果から表示画面のカラーコードを得ています。ここではマスクを操作するだけですので，待避画面を操作する必要はありません。

今回も待避画面やマスクバッファを自前で書き換えたので，終了コードは2「なにもしないで終了」します。「自前でやらなくてもさっきのCOMGVRAM()は使えないのか？」と思う方もおられるかもしれません。確かに，COMGVRAM()ではマスクする操作もあります。しかし，残念ながら「マスクを剥す」という操作はありません。マスクを剥す部分だけ自前でやるというのもあまり意味がありませんし，皆さんに内部構造を理解していただくということでこのようにしています。

(3)SOUND FILER

では，いよいよウィンドウを使った外部ファイルの作成に入りましょう。まずはPICFILERのようなファイルウィンドウを

作ってみます。「おいおい，いきなりかよ」と思われるかもしれませんが，実はファイルウィンドウ自体はEX-WINDOWがサポートしているので，実際はそのあとの処理を記述するだけで済みます（PICFILERなどもこれを利用しています)。ここでのポイントは，

1) ファイルウィンドウの開き方
2) その後の処理

です。

ここでは，OPMやPCMを鳴らすファイラーを作ってみましょう（SOUNDFILER.C)。

ファイルウィンドウを開くには，EXLIB.Hで定義してある構造体WINPTRを渡してやる必要があり，そのメンバは以下のようになっています。

char *title

タイトル名へのポインタです。このポインタで示された文字列がウィンドウのタイトルバーに表示されます。また，この文字列に".EX"拡張子をつけたレジュームファイルを生成しますので，ファイル名に使用できない文字（スペースや¥など）は使用しないでください。

char *ftype1

ファイルウィンドウはPICFILERのように2つのタイプのファイルを扱えますが，そのときにモード切り替えボタンに表示させる文字列を指定します。

char *fname1

ftype1モードのときに，ファイルウィンドウ内に表示させるファイルをワイルドカードで指定します（必ず"*."で始めてください)。

char *ftype2

ftype1と同様ですが，ファイルタイプをひとつしか扱う必要がないときはヌルにしておけばモード切り替えボタンは省略されます（参考：CUTFILER)。

char *fname2

fname1と同様です。ftype2がヌルのときは特に指定する必要はありません。

char *fname

ファイルが選択され，LOADもしくはSAVEボタンが押されたときに，そのファイル名（フルパス）を指すポインタです。

プログラムでは，ファイルタイプ1がOPM，ファイルタイプ2がPCMとしてあります。また，fname1が"*.*"としてあるのは"*.OPM"ファイルだけでなく，"*.ZMS"ファイルなども表示したかったからです（残念ながら複数指定はできません)。これらを指定し，FILEWIN()に渡してやると，ファイルウィンドウが開かれ，結果が戻り値およびfnameに返ってきます（ファイルウィンドウは閉じられます)。

戻り値は，

−1…エスケープキーによってファイルウィンドウが強制終了された。
0…クローズボックスにより終了された。
1…ファイルタイプ1でfnameで示されるファイルのLOADが指定された。もしくはファイルがダブルクリックされた。
2…ファイルタイプ1でfnameで示されるファイルのSAVEが指定された。
3…ファイルタイプ2でfnameで示されるファイルのLOADが指定された。もしくはファイルがダブルクリックされた。
4…ファイルタイプ2でfnameで示されるファイルのSAVEが指定された。

ここではファイルの性質から，LOADでそれぞれを鳴らし，SAVEは無視することにします。鳴らす方法としては，いたって手抜きで，ファイルをそれぞれOPMやPCMにコピーするだけです。当然のことながらOPMやPCMのドライバもしくは相当品を常駐させておく必要があります。

ここで，そのままコピーすると「1個のファイルをコピーしました」というメッセージが表示されてしまいます。EX-WINDOW側でDOSコールPRINTをトラップしてあるので，テキストVRAMを破壊されることはありませんが，いちいち確認ウィンドウをクリックするのは面倒ですし，第一美しくありません。そこで，PRINTをなにもしない関数に割り当て直してあります。

使用法はPICFILERと同じです。が，残念なことに割り込みの関係でしょうか，特にOPMが多少おかしくなることがあるようです。まあ，サンプルということで勘弁してください。

(4)TRIPLE CLICK

今度は実際に自分でウィンドウのデザインをして，ウィンドウマネージャを操作してみましょう。ここでは，ウィンドウマネージャはトリプルクリックまでサポートしているので，それらを使った簡単なゲーム（?）を作ってみましょう（TRIPLE.C)。ここでのポイントは次の2点です。

1) ウィンドウの開き方
2) ウィンドウマネージャの操作法

ルールは簡単，ウィンドウ上のターゲットを左右トリプルクリックを目指してマウスでばしばし叩くだけです。

まずはウィンドウを作らなければなりませんが，これは多少複雑で，EXLIB.Hで定義してある10ワードからなるITEMという

クリックのテスト

アイテムの情報と，さらにそれをひとつのメンバとする，同じくEXLIB.Hで定義してあるITEMPTRという構造体を指定しなければなりません。ITEMはアイテムの機能や座標を示し，アイテムひとつでITEMの配列をひとつ使用しますので，アイテムの数だけ配列を確保する必要があります。ITEMのメンバは順に，

short rev

マウスがアイテム上に重なったときに反転するかどうかを指定し，1のとき反転する，0のときは反転しません。また，254のときはウィンドウの移動をシステムが行い，255のときはウィンドウのクローズとしますが，システムでクローズするわけではありません。クローズに関してはあまり意味はありません。

short ret

ウィンドウマネージャから復帰するタイミングを指定します。0ビットはアイテムをクリックしたあとにボタンを離すまで待つかどうかを指定し，1の場合は待ち，0の場合は待ちません。また，第1ビットはダブル，トリプルクリックを有効にするかどうかのフラグです。0の場合は認識しません。有効にした場合，クリックしたあとの反応が若干遅れますので，むやみに使うべきではありません。

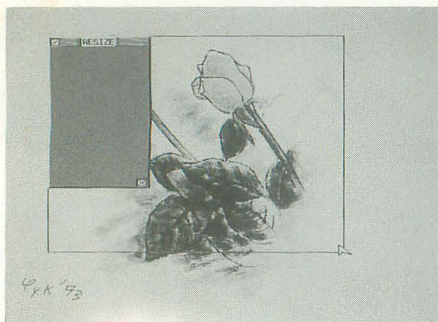
short x1
short y1
short x2
short y2

それぞれウィンドウの左上からのアイテムの左上，右下相対座標です。

short revx1
short revy1
short revx2
short revy2

revを1（または255）にした場合の反転領域の左上・右下相対座標です。

次に，ITEMPTRのメンバですが，

int x0
int y0

ウィンドウのリサイズ

ウィンドウの左上座標です。ただし，−1としてウィンドウをオープンした場合は，それ以前に開いていたウィンドウと左上隅をあわせて開き，これらの変数にはその座標が返ります。また，ITEMのrevを254としてシステムでウィンドウの移動をさせた場合も同様にこれらの変数に移動した先の左上座標が代入されます。

int x

int y

ウィンドウのXYサイズです。

int n

アイテムの数です。

ITEM *i

前述のITEMへのポインタです。

これらの変数はユーザープログラムでは気づかないうちにシステムによって書き換えられたりすることがありますので，グローバルで宣言しておいたほうが無難でしょう。

さて，ウィンドウ情報の指定が済んだら，まずその情報をWINITEM()によりシステムに転送してやります。そのあとタイトルを引数としてOPENWIN()を呼びますが，それだけではタイトルバーだけののっぺらぼーなウィンドウしか開きませんので，自分でアイテムなどを書き込んでやらなければなりません。

プログラムではWINBOX()を用いて，へこんだようなボックスでターゲットを書いています。また，ウィンドウの色はEXLIB.HでWHITE,GRAY1,GRAY2,BLACKのマクロで定義してあります。

先ほどはいいませんでしたが，EX-WINDOWから外部ファイルを起動した時点ではマウスカーソルは時計（BUSY）になっています。したがって，ウィンドウなどを扱うときにはマウスを矢印（READY）にしておきます。MCSET()を使用すればマウス形状を変えることができ，引数0で矢印，1で時計，2でスポイトの形状になります。状況に応じて形状を選択してください。

さて，ウィンドウを書き終わったらウィンドウマネージャMANAGE()の出番です。引数が1の場合はアイテムのどれかが選択されるまで待ちますが，0では待ちませんので，同時に別の処理を並行させる場合などに用いてください。

戻り値は以下のようなビット構成になっています。

000 F L R L R NNNNNNNN

F

引数を0とした場合，マウスがアイテム上にないときのフラグです。このとき，マウスのクリックは認識しません。

L L

左ボタンの押下状態です。以前のバージョンとの互換性の問題から飛び飛びのビットになってしまいました。00でOff，01でシングルクリック，10でダブルクリック，11でトリプルクリックです。

R R

右ボタンの押下状態です。

NNNNNNNN

選択されたアイテムの番号です。ただし，255のときはエスケープキーが押されたことを示します。

エスケープキーが押された場合とクローズボックスをクリックされた場合はCLOSEWIN()によりウィンドウをクローズして終了し，ターゲットをクリックした場合はchk()関数で判別，表示しています。こちらの関数の説明は特に必要ないでしょう。

「ウィンドウを描く」といっても，このくらいならEX-WINDOWがサポートしてくれるので，さほど難しくもないでしょう。これがサポートしていないことをしようとすると，かなり面倒なことになるんですが……。

(5)RESIZE WINDOW

で，次はその面倒なことをやってみましょう。ウィンドウシステムではたいていやっているウィンドウのリサイズ，なんとEX-WINDOWではサポートされていません。サポートされていないものをどうやってやるかというと，それはもう「力技」で「真面目」にやるしかないわけで，別にこんなところでそんなことしなくてもいいような気もしますが，適当に説明しちゃいますので，ソースをしっかり読んで理解してください（RESIZE.C）。ここでのポイントは，あんまりない，かな？

強いていえば慣れてください。慣れてしまえばどってこたぁありません。では始めましょう。

ウィンドウを書くところまではいいでしょう。もう説明しませんので，各自で理解してください。いきなりRESIZE()関数から説明します。さて，この関数の頭でマウスの座標から，なにやらdx,dyなるものを求めています。itemptr.x0はウィンドウの左端の座標で，itemptr.xはウィンドウの幅ですから，その和はウィンドウ右端の座標ということになります。

それからマウスのX座標を引くと，ようするにマウスからウィンドウ右端までの距離を求めているんですね。dyについても同様です。で，IOCSコールMS_LIMITによってウィンドウが画面からはみださないようにマウスの移動範囲を限定しています。SX-WINDOWなどではウィンドウは画面からはみだしてもいいようになっていますが，EX-WINDOWでは真面目にクリッピングしていないので，右からはみでたものは左から出てきてしまい，下からはみでたものは上から出て……こないで，バスエラーになってしまいます。

ちなみにMS_LIMIT()は先頭座標と最終座標に同じ値を指定できないので，EX-WINDOW側で手を加えたものに差し替えてあります。そのあと，ドラッグしている間はマウスに合わせてREVBOX()で枠を書き続けています。

サイズが決まったら，ウィンドウをクローズするのですが，ここでは余計な部分だけ消すようにしました。全部消してからまた書き直してもいいのですが，このほうが美しく，正道でしょう。ウィンドウを消すにはBUF2GRAM()を使います。名前のとおり，待避画面からG-RAMへコピーする関数です。

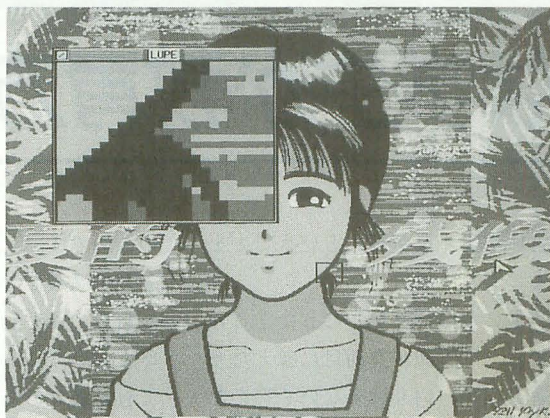
このように余計な部分を消したあとに，ウィンドウ情報をウィンドウサイズにあわせて書き直し，再びウィンドウを開くのですが，その前に必ずWINITEM()でウィンドウ情報を転送し直してください（いちばん長いサンプルだったのに，こんなに簡単に終わってよかったのだろうか？）。

(6)LUPE

ここまで理解していただけたでしょうか。今度はいままでの応用としてルーペを作ってみましょう（LUPE.C）。ただ拡大して見るだけですが，ここまで理解できればほぼ完璧でしょう。では，プログラムを見てください。

この外部ファイルは，マウスでウィンドウ外をクリックすると，その近傍24×24ドットの領域がウィンドウ上に8倍に表示されるようになっています。

ここで，拡大させる領域を指定するのに

ウィンドウの外を認識させてやる必要があることがおわかりいただけると思います。MANAGE()の引数に0を指定して，マウスがアイテム上にないときのフラグを利用する方法もありますが，そうするとそのあとが面倒なので，ここでは画面全体をひとつのアイテムにしてしまうという方法をとりました。ウィンドウマネージャはウィンドウアイテム上にマウスがあるかどうかを順に判別し，あればそれ以降のアイテムは無視しますので，リストのようにウィンドウ内とウィンドウ外のアイテムを渡してやればよいのです。

ルーペ

外部ファイルの実行

先ほどもいったとおりEX-WINDOWはウィンドウが画面からはみだすのを認めませんので，ウィンドウ外の相対座標を(−511,−511)−(511,511)とすれば十分であることはおわかりいただけるでしょう。

こうしておいて，ウィンドウ内を指定された場合はなにもせず，ウィンドウ外の場合だけREVBOX()で枠を移動させ，ウィンドウに拡大しています。ここで，このプログラムではマスクを無視してその下を表示するようにしてあります。また，今回はルーペの枠の表示の都合で，ウィンドウの移動を自前でやるようにしてあります。

ウィンドウの移動の方法はいたって簡単で，MOVEWIN()に引数として現在のマウスの座標を渡してやるだけです。マウスのドラッグにあわせてウィンドウが移動し，移動後のウィンドウ左上座標がitemptr.x0とy0に返されます。

このプログラムではアイテム2にLUPEアイテムを設定してあるにも関わらず，なにもしていません。ここにエディット機能をつけたり，右クリックでスポイト機能をつけたりすれば，それで立派な外部ファイルとして実用になると思います。あとは皆さんで改良を加えてみてください。

(7)EXECUTE

最後にちょっと毛色の違う外部ファイルを紹介しましょう。外部ファイルからさらに任意の実行ファイルを起動します。ここでのポイントは次の2点です。

1)　キーからの入力
2)　割り込みの停止

ポイントといっても，EXコールを呼び出すだけです。まず，キーからの入力ですが，EXコールにはライン編集入力ファンクションが用意されています。LEDIT()という関数ですがこの関数の引数は，

```
LEDIT( x, y, disp, max, str );
int x, y, disp, max;
char *str;
```

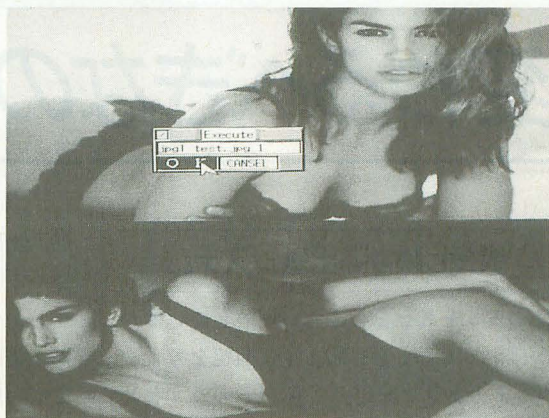
となっています。

ここで，x,yというのは文字列を表示する左上x,y座標で，dispは画面に表示する文字数，maxは編集可能な最大文字数です。そして，strは編集文字列へのポインタで，LEDITで文字列を編集するとこのポインタの示す領域に反映されます。入力時にはカーソルキーやDEL,BSキーが使用でき(常にインサートモードです)，カーソルキーの上で文字列の先頭へ，下で最後へカーソルが飛びます。確定はリターン，ESCでキャンセルし，それまでに編集した内容は破棄されます。

割り込みの停止というのは，どんなファイルが実行されるか予期できないためです。これもEXコールに用意されており，MINT()という関数を呼ぶだけで，引数0で割り込み禁止，1で許可します。割り込みの禁止によりマスクの点滅が停止し，マウスカーソルが表示されなくなります。

これだけいってしまうと，プログラムの説明もあまり必要ないでしょうが，exec関数については少し触れておきましょう。

まず，割り込み禁止にするのはいいとして，printf()関数で文字列を表示しています。といっても画面には直接表示されず，3月号で説明したLOGファイルに落とされます。これはまあ，特に意味はありませんが，あとでなにを実行したかわかるようにというのと，実行ファイルがなんらかのメッセージを発したときに，あとでLOGファイルを見て誰がそのメッセージを出したかわかるようにするためです。

そのあとファイルをフラッシュし，system()関数でstrの中の文字列をHuman68Kに渡しています(system()を使っていますのでcommand.Xの内部コマンドも呼べます)。外部ファイルが終了すると，画面モードがどうなっているのかわかりませんので，to64k()という関数で65536色モードに変換しています(この辺はAMIEXのコマンド実行でもやっています)。

to64k()という関数はto64k.sというファイルに入っていますが，そちらの説明は省かせていただきます。最後に割り込みを許可して，終わりです。まあ，この外部ファイルはLEDIT()のサンプルとして作ったものですので，それさえわかってもらえればいいでしょう。ただし，あまり全角文字に真面目に対応していませんので，全角文字を使用する際には注意が必要です。入力してできないことはありませんが。

最後に

Z's-EX ver.1.0のシステムは閉じていました。拡張しようとすれば，Z's-EX自体のソースに手を加えてコンパイルし直すしかありませんでした。ver.1.1になって外部ファイルが扱えるようになり，その気になればなんでもできるようになったとはいえ，システムのサポートが貧弱で，プログラマに多くの負担がかかっていました。

いま，Z's-EXはver.2.0となり，それとともにMATIER-EXも発表され，システムのサポートも十分とはいえないまでもかなり強化されました。3月号ディスクに収録された外部ファイルを見てもそれはわかっていただけると思います。

あとはこれらの外部ファイルを参考に，皆さんで環境を構築し，オリジナルのEX-WINDOWを作り上げてください。EX-WINDOWはそれが可能なシステムなのですから。

変わるべきなのは学生か教育方法論か，

成長し続ける出題システム

あることXをどのくらい理解しているかを測定することを目的とする問題を考えます。そのような問題のうち，(どのようなXに対してもまったく同じ問題ですむという意味で)普遍的であるという特長をもつ,次のような問題はどうでしょうか？

> **問題A　あることXをどのくらい理解しているかよく測定できるような問題を作れ**

Xをよく理解していないと，この問題Aによく答えられない，つまり適切な問題は作れないでしょうし，あまり理解していないと，よくない問題を作ってしまうでしょう。よくない問題というのは，たとえば，簡単すぎる問題では全員満点ということで駄目です。また，もちろん難しすぎても駄目です。ごくわずかな人にしか解けないような問題で，そのほか大多数の人がみな0点となってしまってはまずいからです。

ただし，上に挙げた問題Aも決して万能というわけではありません。この問題Aの解答として，

> **解答A　あることXをどのくらい理解しているかよく測定できるような問題を作れ**

というものを返される可能性があるからです。

もし，このひねくれた解答に対して×をつけたとします。するとそれは，解答A「あることXをどのくらい理解しているかよく測定できるような問題を作れ」が，あることXをどのくらい理解しているかをよく測定できない悪い問題である，ということになります。その場合，それと同一の，問題A「あることXをどのくらい理解しているかよく測定できるような問題を作れ」も同様に悪い問題であるということになり，結局，出題自体がよくなかったということを認めることになります。

しかし，逆にこの解答を○にしたところで，実際には，解答者が真面目に考えたうえでの解答なのか，何も考えないで答えたのかはまるでわかりません(多少機転のきく奴かな，ということはわかりますけれど)。したがって，結局は，問題Aは解答者の理解度をはかることができなかった悪い問題ということになります。

そこで，しかたなく，そういう解答はあらかじめ封じておくことになります。

> **問題　あることXをどのくらい理解しているかよく測定できるような問題を作れ。ただし，「あることXをどのくらい理解しているかよく測定できるような問題を作れ」的な解答は認めない**

これで，解答する人がきちんと自分の頭でよく考えてくれて，しかも，その理解度をはかれる問題となったのでしょうか？いまいち自信がありません。でも，この出題には大きな長所があります。それは，戻ってきた解答のなかに，このような類の問題ではない，もっと率直で良質の問題が戻ってくる可能性があるということです。そうすればしめたものです。次回に出題する機会に，この出題と併用して，その良質な問題も出せばよいからです。これで，理解度をより正確にはかれるように成長し続けるシステムが完成するわけです。

ボトムアップなやりかた

先ほど示した問題でXのところをCに代えたものが，実は僕が担当している，C言語を習得する講義(演習)の最後に出したレポート問題のうちのひとつです。この演習は工学部の2年生がそれぞれワークステーションの前に座って,(今年度から始まった講義でしたので)毎週僕がヒーヒーいいながら作った教材に従って実際にCのプログラムを入力しながら学んでいくという内容のものです。

どの講義でも同じだと思いますが，学生が自分の頭を働かせてその講義に参加しているということが，その講義が学生にとって有意義であるための必須条件でしょう。その点，この講義は，1人1台ずつのワークステーションという環境が与えられ，しかも個人個人がその理解速度に応じて消化していける教材も与えられているのですから，比較的恵まれた条件であるということができるでしょう。

一般に，あることを教える方法として，具体的な例から入って体で覚える，というところから始めるボトムアップなやりかたもありますし，その逆に，高尚な理論や普遍的な概念から入るやりかたもあります。この授業では，せっかくワークステーションが目の前にごろごろころがっているのですから，前者のやりかた，徹底したボトムアップのアプローチで体で覚えてもらうことにしました。

Cにはこういうデータ構造があって……とか，こういう場合には構文が用意されていて……などとまず説明して，それからそれを実際にプログラムを走らせてみて確認させるというのではなく，まず，わけもわからずでいいから，とにかく自分で入力して走らせてみて，それから，うまくいったのはなぜか，うまくいかなかった理由は，などと考えさせることにしたのです。

たとえば，毎週作った教材は，まず簡単なプログラムを入力してコンパイルし，実行させてみろという「指令1」から始まります。1年生のときに，エディタの使いかたやメールの出しかたなどの基本的な計算機リテラシーの教育は受けているので，導入は比較的楽です。そして，一切の説明なしに「指令2」に進みます。

「指令2」では，指令1で入力したプログラムに対していいかげんな文字を付け加えたり，文字を消去させたりしたあとで，コンパイルしてみて，エラーを出させるというものです。これにより，コンパイルしたと

「情報」の時代に？

きに，エラーが出てきても冷静でいられるように作ってみました。もちろん，ずらずらと出てきたエラーメッセージをすべて理解できるようになれというのは，すぐには無理な話ですけれど。

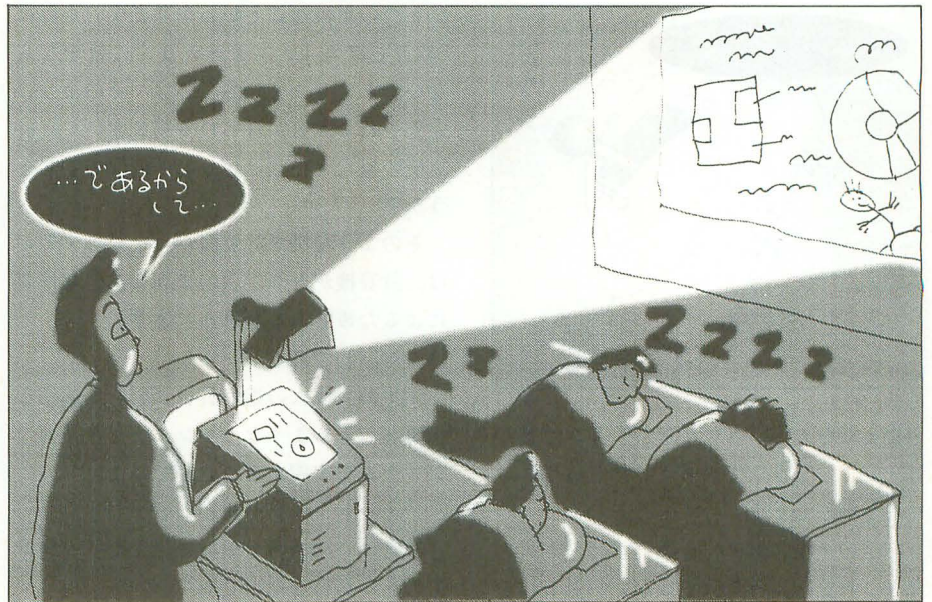
教材の各章の終わりには，きわめて簡単なまとめが「記憶」として入っていて，初心者が間違えそうなミスは「駄目」という項目のところに挙げられています。そして最後は「問題」です。

毎回この教材が配られ，学生は，そこに含まれる「指令」と「問題」の解答として作ったプログラムやその実行結果などを，次回の講義までにレポートで提出します。このレポートは電子メールで送ります。レポート管理システムは，提出期限の時刻などを設定すると，自動的に受け取りの記録を残したり，課題ごとにまとめたり，レポート提出者には受け取り（提出が遅れているときにはその注意も含まれる）の返事を出したりしてくれるので，きわめて便利です。

まあ，とにかく，まずは体で覚えてもらって，その後，何とか頭で理解するようにスムーズにつなげようと，多少の努力をしてきました。そして最後のしめくくりのレポート課題のうちのひとつが最初に挙げた問題だったわけです。

他人のレポートをコピーして終わりとはならないように，解答がひとつではないようにして自分の頭で考えるように，また，こちらも毎回悩まないで合理的に出題するために次回の問題のネタも集まるようになどと考えたわけです。

参考までに，最後のレポート課題として出したほかの課題のうち2つほど挙げてみましょう。

問題1　本計算機システムのccコマンドでコンパイル中に発見されるエラーに関して，どのようなエラーでどのようなメッセージが表示されるかを自分の経験に基づいてまとめよ

illustration：Haruhisa Yamada

問題2　C言語の習得において最も理解が難しかったところを具体的に述べよ。さらに，もし自分と同じ習熟度に達している他人にその部分を説明するとしたらどうするか，自分の言葉を用いて書け

2つの小さな失敗例

この例は，計算機のプログラミング言語を学ぶというケースでしたので，その講義において計算機を使わせるというのはきわめて自然なことでした。ここで，学ぶ対象としての計算機ではなくて，手段やメディアとしての計算機の使いかたということに視野を広げてみます。

計算機に関係しない講義，あるいはもっと広く教育一般の場で，計算機やいわゆる「マルチメディア」を使うという考えはそう新しいものではないでしょうし，実際，すでにいろいろな形で実施されているわけです。しかし，うまく利用するということはきわめて難しいように思えます。小さな例を2つほど挙げましょう。

失敗例1

ある授業で，オーバーヘッドプロジェクタ（OHP：スクリーンに透明シート上の絵を投影する装置）とMacintoshとプロジェクションパネル（パソコンの画面出力をOHPシートの代わりにする装置）を使ってみたことがあります。

その授業の終わりに，感想を聞いてみたところ，投影されている文字や図をノートに書き写すのは目が痛くなっていやだ，という学生が半分以上いたのです（まあ，これは僕のやりかたの未熟さにも起因するのでしょうが），残念なことに！　それで，その次からは黒板に書くごく平凡な授業に戻ってしまいました。

失敗例2

研究室で英語で書かれた文献を輪講するゼミにおいて，原文を日本語に訳してそれをほかの参加者に配るという形式をとっていたときの話です。最終的に全員がワープロを使ってきれいに出力したものを配ったのですが，そういう形式をとらなかったとき（日本語訳の配布を義務づけなかったとき）に比べて，学生の理解度が明らかに低く感じられました。要するに，機械的に日本語に逐語訳してプリンタできれいに印字すると，何となく「ああ，わかった。ひと仕

第79回　知能機械概論――お茶目な計算機たち――

変わるべきなのは学生か教育方法論か，「情報」の時代に？

事終えたな」という気になり，そこで思考停止してしまう学生が多くいたということになるのでしょう。

以上の2つの例は，まあそれほど本質的なところを物語った例ではないかもしれませんが，いずれにせよ，計算機や新しいメディアが教育の現場に入れば，それで自動的に何かがよくなるのだと期待してはいけないということを身にしみて感じているということがいいたいのです。

教育とはコミュニケーションだ

計算機やマルチメディアなどが教育の現場に入ること，それによってどういう効果があるのか，まずそれをきちんと考えなければいけないのでしょう。面白そうだからとか，学生が喜びそうだからというだけでは駄目なのです。

まず，基本的なところから僕の考えを述べることにしましょう。先に述べたことの繰り返しになりますが，学生が自分の頭を使うこと，これは必須条件です。そして，教官側，これにはもしかしたら計算機そのものも含まれる場合もおおいにあると思いますが，そちら側は，メッセージ(それはもちろん情報であるわけですが)を学生側に与えるわけです。そして，それに応じて学生側が，何らかのメッセージを教官側に返すということが必要でしょう。もちろん，メッセージを返すためには当然頭を使う必要があります。

教育というものをこのようにとらえると，ひとつの単純なことがわかります。それは「教育とはコミュニケーションにほかならない」ということです。では，そのように定義される教育が行われる場において，計算機の果たす役割はいったい何なのでしょうか？

そのような場で計算機が力を発揮するのは，計算機がもつ，次のような重要な性質によるためであると思われます。

1)　対話的である

学生個人個人の進み具合に応じて適切な指導ができるような，インタラクティブなシステムを実現することができる。

2)　ネットワークを組むことができる

教官からの情報を全員に伝えたり，各学生の進み具合を把握しやすい。

3)　計算能力が高い

ある事柄を計算機内で実際にシミュレートして，学生の入力に対してその結果をリアルなグラフィック出力で見せることができる。

以上のような計算機の特長を活かすには，具体的にどのように教育の方法論を作ればいいのでしょうか？　忘れてならないのは，あくまでも教官側と学生との間のコミュニケーションをより豊かにするためにこそ，計算機あるいはマルチメディアを使わなければならないということです。

たとえば，計算機による鮮やかな画面表示だけを売り物にしたようなシステムを作ると，それは学生の興味を集める結果にはなるでしょうが，いちばん重要なこのコミュニケーションというものに対しては何の効果もない，場合によってはむしろ障害になってしまうという結果をもたらすこともありうるということです。

いくらきれいな「視聴覚機器」を使ったとしても，それが教官側から学生側へのメッセージを補強するだけのものならば，学生から教官側へのメッセージの流れはむしろ押しつぶすことになってしまう場合が少なくないのではないかということが，気をつけなければならない重要なことであると僕は思っています。

ああ面白い，きれいだ，ふふーん，とかいって納得した気になっても，時間がたったあとで，そのきれいだったなどという余韻だけが残り，実際に何がメッセージだったのかということがすっぽり抜けてしまっていては，まったく意味がないというわけです。

そうではなく，学生それぞれの反応，たとえばこれを試したらどうなるかな，というような気持ちが起こったときに，即座にそれを計算機に伝えてシミュレーションが行われるようなシステムならば，そこには，きわめて豊かなコミュニケーションが成立するのです。

若いジェネレーションとメディア

実体もはっきりしないうちに垢にまみれた言葉のひとつに「情報化社会」というのがありますが，実際，この情報化社会とやらのおかげで，人の感覚やものの考えかた，価値観などが確実に変化しつつあると思います。そして，若いジェネレーション(わしゃ，じじいか？)はその変革の主人公そのものです。したがって，彼らがその時間の多くを費やす教育の現場も急速に変わらねば駄目に決まっています。

厚顔無恥という非難を承知でいわせてもらえば，学生が悪くなっていままでの教育方法に従わなくなってきたというのは間違いであり，単に，「情報」というキーワードで語られる社会の変化に適応する新しいタイプの人間に，教育のほうがついていけないということなのでしょう。そして，教育の方法論すらまだ確立されていないのでしょう。

厚顔無恥と書きましたが，決して軽い気持ちでこのようなことを書いているのではありません。若いジェネレーションとは，すなわち未来そのものを表しているのです。そしてまた，教育という問題は，僕自身にずしりとのしかかってくる問題でもあるのですから。

愛読者プレゼント

1

ファミリーソフト ☎03(3924)5435

マッドストーカーX68

X68000用 5"2HD版 7,800円(税別) 3名

ゲームバランスのよさやロボットの美しいアクションが好評の格闘ゲーム。レビューを担当した瀧氏絶賛の品です。しばらくX68000は御無沙汰だったファミリーソフトですが，次回作も開発中です。

2

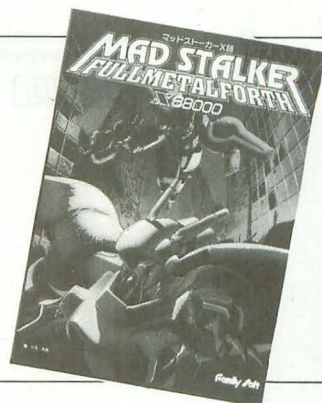
サクセス ☎03(3791)2820

キーパー

3名

X68000用 5"2HD

8,800円(税別)

ログインソフトウェアコンテストのグランプリ受賞作をもとにしたパズルゲーム。石板を押したり引いたり，だけどほっとくと居眠りしちゃうプクルとピクルが可愛いのだ。

3

CONNECTLINE ☎0899(26)7821

デジタルアートコレクション

A vol.1
B vol.3
C vol.5
D vol.7 各5名

X68000用
5"2HD版

各1,500円(税別)

アマチュア人気CG作家の作品集。先月号の「SOFTWARE INFORMATION」に何点か掲載されています。4096色中16色使用の作品を収録したパッケージ版をプレゼント。希望賞品の記号を明記してね。

4

ストリートファイターIIダッシュグッズ

日本全国で熱い闘いが繰り広げられていますが，さらに3度くらい気温上昇を招くといわれる(?)グッズをプレゼント。勝負だけじゃなく，希望賞品記号の明記も忘れずにね。

A CPSファイター
(スーパーファミコン/ファミコン用)
9,800円(税別) 3名

B シャープSTAFFジャンバー
非売品 5名

C メガホン
非売品 5名

D ストIIイベントフラッグ
非売品 2名

E マーカーセット
非売品 8名

5

TROUBADOUR RECORD

CD「マニキュア団」

2名

知る人ぞ知る「まにきゅあ団」。正統派大衆テクノポップ楽団。メンバーは，あの細江慎治をはじめ，ナムコのゲームサウンドデザイナー。そのまにきゅあ団友の会会員証つきシングルCDを2名様に。江口響子さんからのプレゼントだよん。

プレゼントの応募方法

とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1994年4月18日の到着分までとします。当選者の発表は1994年6月号で行います。また，雑誌公正競争規約の定めにより，当選された方はこの号のほかの懸賞には当選できない場合がありますので，ご了承ください。

2月号プレゼント当選者

1 X68000ログインソフトウェアコンテスト傑作ゲーム選 (愛知県)佐々木恵司 (京都府)鶴本武浩 (大阪府)陣山達夫 2 マウス (北海道)山本晃大 (秋田県)小名 司 (岐阜県)大熊 龍 (敬称略)

2月号モニタ当選者

A OS-9ソフトウェアパッケージセット (栃木県)中内英裕 B Y300-A ver1.1 (愛知県)岡部和秀 (敬称略)

以上の方々が当選しました。商品は順次発送いたしますが，入荷状況などにより遅れる場合もあります。

石の言葉，言葉の夢

[第2回]

コードレス

Ogikubo Kei
荻窪 圭

3DOプレイヤーが1週間だけうちにきた。わはは。

3DOって，かの有名なマルチメディアプレイヤーの規格で，アメリカでは鳴かず飛ばず状態だって噂だけど，日本では大々的にパナソニックがキャンペーンはってるみたいで，ある程度はきそうな感じもある。でも，製品が出たとたん，「3DO」「3DO」って叫ぶ声もなくなっちゃったみたいだし，どうなのかしら。値段も7万円以上。FM TOWNS MARTYよりは安いけど。(編集部注：この後，松下電器産業の3DOゲーム機「リアル」の定価は54,800円に値下げが発表されました)

まあ，面白い製品ではあるわな。英語版を遊んでみたんだけど，ソフト作りが難しそうだ。ROMカートリッジと違って，CD-ROMはロードに時間がかかるから，凄く鬱陶しいし，設計が悪いと，無意味な待ち時間に悩まされる。それでも，付属する「CRASH'N BURN」は制覇したけどさ，完成度はいまいちだったし。そうだな。3DO用に「バーチャファイター」でも出たら買ってしまうかもしれない。

と，このまま家庭用マルチメディアプレイヤーの話をしてもいいのだが，これはほかのところに書くから，ここではしない。申し訳ない。ま，そのうちに，やります。

今回は，コードレスの話。

子機モデムってどうかしらん

パソコン通信をするとしよう。

私なんかは狭い家に住んでいるわけだし，電話回線を自分の好きなように使っているからいいんだけど，家族がいたりすると困ってしまうわけだ。自分の部屋まで電話線をひっぱっていくのはけっこう難儀。

それに，コードレスホンがこれだけ出回ると，電話しようかなと子機を手に取ったら，「ピーガガガガ」ってなんじゃこれ，ってことになる。逆に，別の部屋で子機を使っているときに，通信しようとして迷惑かけたり。顰蹙(ひんしゅく)ものだ。そういえば「子機」って口に出すとヘンだよね。「コキ」って，とても電話のイメージじゃない。

そこで，考えたわけだ。その名も「子機モデム」。凄いよ，これは。シャープでも製品化しないかな。

発想は，どうしてコードレスホンの子機が電話機じゃなきゃいけないか！ってところにある。だって，そうだよね。

コードレスホンにいくつか無線でぶらさがっている子機のひとつがモデムだ，っていうそれだけのことだけど。

まず，「子機モデム」の通信先は親機である。親機といっても，メーカーによって全然違うわけで，子機モデムには設定スイッチがついている。シャープ，ソニー，ケンウッド，ビクター，NTTなどなど主要電話機メーカーの主要機種にすべて対応しており，設定スイッチでもって，親機がどのメーカーの製品かを判別するわけ。

で，子機モデムが作動すると，親機は子機がアクティブになったのを検知し，通常の子機に対する動作と同じことをする。それだけである。

子機モデムは，アナログ信号に変換したデータを，電話回線ではなく，コードレスで親機に流す。親機がそれを電話回線にそのまま流すわけで，子機モデムがデータを受け取るときはその逆。

つまり，汎用の子機モデムをパソコンにつなぐとだな，コードレスホンの親機さえあれば，家中どこででも通信ができるわけだ。

便利でしょ？

しかも，コードレスホンとしての機能も十分にもっているのだ。内線呼び出しとか，パソコン側から留守電をセットしたり留守電の内容を聞いたりという機能。子機モデムとパソコンを組み合わせれば，留守電の内容をハードディスクにそのままサンプリングして，必要な事項をスケジューラに貼り付けたりメモに残したり，オプションの音声解析ソフト「聞き取りくん」を使えば，音声を解析して留守電の内容をテキストファイルにしてくれるってんだから，こいつは凄い。子機モデムが普及するにつれて，対応ソフトも続々登場する。

当然パソコン側にはそれ相応の音声機能が必要だけど。それはまあ，あって当然やね。22kHzのサンプリング機能くらいは。

留守電の応答メッセージもパソコンで作れるから嬉しい。BGMを組み合わせたり，音声合成ソフトでテキストを読み上げさせたり（これは，若い女の子がイタズラ電

話防止に使ったりもするのだ)。

姉妹品に，子機アダプタがある。これは，通常のモデムのLINEジャックにつなぐ小型の装置で，これがあれば，手元のどんなモデムでも子機モデムとして使えるわけ。でも，コードレスホン機能はないから，親機との高度な通信はできない。そこが残念なところだけど，ノートブックパソコンの内蔵モデムにこいつを装着すれば，家中どこにいても通信できるから便利だ。

シャープさん，出さないかね。子機FAXモデム。データ通信は14400bpsのV32.bis。FAXも同様にclass2。10社の小電力型コードレスホンに対応していて，値段は39,800円(無理かしら？)。ソフトウェアはWindowsキットとMacintoshキットとX68000キットがあって，こちらは9,800円。どうだどうだ。

子機モデムからコードレスLANへ

子機モデムがあるとなると，子機モデムをいくつも用意して，子機モデム同士の内線通信なんかしたいよね，ね。

そもそもさ，無線LANなんていうから，話がでかくなっちゃうわけで，クライアント/サーバーコンピューティングとかいうから難しそうに聞こえるわけで，家庭内なんだから話はもっとパーソナルでなきゃいかんのだ。もう，「コードレスLAN」と「内線通信」でいい。

子機同士で通信する。LANも組める。そうなると，相手はパソコンでなくてもいいわけだ。

子機PDAもほしいな。PDAって，アップルのNewtonだとか（そういえば，シャープが作っている日本語版はどうなったのかしら)，カシオのZoomerだとか，AT&Tもなんか出すみたいだけど，そういうもの。パーソナル・デジタル・アシスタント。電子手帳とペンコンピュータの間に位置するもので，賢くて，通信機能がしっかりしている，と。

これに入っている情報って，パソコンにバックアップがほしいし，パソコンと整合性をもたせて，パソコンから必要な情報をロードしてもっていきたいと思うんだけど，これが面倒。電子手帳使った人ならわかると思うけど，家に帰って，ケーブルつないで，パソコン側のソフト起動して，って，結局，やらなくなるのだ。

そこで，子機PDA。家に帰り，PDAアダプタに装着するだけで，自動的にコードレスLANからパソコンのほうへ信号がいって，専用のソフトがバックグラウンドで動きだし，データの転送をやってくれるの。いいよね。このアダプタはバッテリチャージャーも兼ねているから，家に帰って即装着。その後のコントロールはパソコンから。

子機PDAに続くは，子機FAX。子機ビデオ。子機テレビ。子機エアコン。凄いね，これは。

最後はコードレス本。NECのデジタルブックみたいなしょうもないのではなく，もっとちゃんとしたやつね。コードレスで読みたい本をロードして，持って歩くの。子機PDAとどこが違うのか，っていうと，製品名だ。ただそれだけ。

そういえば，どこだったかな，赤外線リモコンをもったコードレスホンの子機があったな。あれ，笑えた。確かに，子機だったら絶対になくすことはないから，うちにはいいかもしれない。今度使ってみたいのである。

子機同士のコードレスLANとなると，実のところ大きなネックがある。子機と親機の間って，要するに，音声がそのまま電波にのって飛び交っているだけだから，回線のクオリティってやつを考えると不安なものがあり，LANなんていうくらいだから，9600bpsとか14400bpsでは心許ないわけで，やっぱり，親機と子機の間をそれなりのデジタル回線で結んでほしいな，と。

だから，第一段階は14400bpsで，簡単なメッセージのやりとりだろうね。普通にLANするには遅すぎる。14400bpsっていえば，秒1.8Kバイトだからもう話にならない。最低でもその10倍，できたら100倍はほしいところだ。

というわけで，そのうち，デジタルコードレスホンが出る。勝手に決めた。出るのだ。親機と子機の間をデジタルで結ぶってわけで，親機をISDNにつなげば，もう，フルデジタルで，幸せ。そうなれば，ほかのLAN(たとえば，会社とか)ともリモートアクセスして快適なISDN生活ってわけだ。

あまり度が過ぎると，「支援」でなくて「管理」になっちゃうから，この辺で止めておこう。規格を考えるのも大変そうだし。

ほんとにできるのかな

と，勝手に書いてきたわけだが，ふと思うのだ。コードレスホンの子機って，こんなに何でもつないでいいのかしら。通信機器って，郵政省だかの認可が必要なのだけど，こういうケースはどうなるのかしら。勝手にコードレスホンのシステムを応用しちゃっていいのかしら。こーゆー使い方をしていて，小電力コードレスホンの通信チャンネルは不足しないのかしら。

うーん。ちょっと不安になってきた。でもいいや。

かくして，コードレスホンと家庭内AVシステムとパソコンはうまく結合したのであった。しまいには，電話機だか，小電力コードレスホンのチャンネルを略奪した無線LANだかわかんなくなってしまったけれども，まあ，めでたしめでたし，ってことにしておきましょうね，ね。

ヘロンの聖水販売機

Takazawa Kyoko
高沢 恭子

機械の発明や発達は「楽をしたい」という欲求から始まります。そして「より便利に」「効率よく」などと要求は高まり，その一方では機械を騙す工夫さえ出てきます。過去にも，そして現在でも……。

地下鉄日本橋駅の改札口を出たとき，ちょうど自動改札機の1台がメンテナンスの最中だった。いそぎの用事がいくつかあったのについ見学してしまったのは，前日，自動改札機のフシギについて家族と話をしたばかりだったこともある。

乗り継ぎは苦手？

乗降客は多いが，整備中の改札機だけは避けて通るから見学のスペースはある。
「けっこうコレね，故障が多いんですよ」
修理をしていた技術者が，私がたずねる前に話してくれた。
「定期的に検査もするんですけどね，やっぱりトラブルで駆けつけるほうが多い感じですよ」
大手の電機メーカーが3社ほど機械を出していて，メンテナンスの会社はどの機種でも手がけるのだそうだ。
「CPUは286クラスのものが入ってます。このベルトはネ，切符をヨコに入れる人や裏がえしに入れる人がいるでしょ，どんな入れかたをしても内容を読み取れるような修正の働きをするんです」
1台の機械が月に1度くらいはトラブルをおこすのだそうで，この機械からも，飲み込んでいたらしい切符が何枚か取り出されていた。

自動改札機はそのほかにも手の込んだしかけがある。扉の開閉は遊園地の設備のように面白い。回数券を入れると駅名が印刷されるし，子供の切符で通過するとセンサーが働いて，おとなとの区別を表示する。あまりいろいろな能力を見せられると，この機械がどこまでやりとげるだろうと試してみたくなることもある。トオルの友人D君は，切符をクシャクシャに丸めたあと，それを広げて投入口に入れてみた。これにはさすがの機械も停止してしまうことがわかったが，取り出された切符が異様にシワだらけだったために，D君はきびしく注意を受け，しかも罰として何割増しかの運賃を請求された。

自動改札機は，正確で完璧なチェックをする機械のはずだが，不可解だったこともある。

三が日のある日，夫と外出しての帰りだった。小田急線から地下鉄千代田線に乗り継ぎ，さらに東西線で自宅のある下車駅まできた。もともと地下鉄と私鉄との連絡や，同じ地下鉄でも営団と都営との連絡は，切符をもとめるとき注意深くなければならない。小田急線の和泉多摩川駅と自宅のある東西線の駅との間は，自販機で買える最高金額の切符では料金が不足で，精算しなければ改札口を出られない。

ところがその日は，和泉多摩川駅の駅員が自販機で買える金額の「330円でよい」と教えたのもヘンだった。私たちは2人分の切符をその料金で買い，下車駅まできて自動改札機に切符を入れた。夫はなぜか通過できたが，私は警報音とともに行く手を遮断された。それが正しい判定だったのだろうが，ともかく同じ金額の乗車券で一方だけ精算となった。

左利きの人で，体の右側に機械があるのが不便だという人もいる。じっさいに左手でウッカリ隣の機械に切符を入れてしまう人もいるそうだが，便利も多数決で取り入れるのだからしかたない点もある。いまのところこの改札機は，右利きで，荷物が少なくて，できれば定期券でない人に，いちばん便利なのだ。

パチンコ玉のヒント

用事を終えてふたたび日本橋から乗った地下鉄の車内で，メンテナンス会社の社員らしい人たちといっしょになった。3人がそれぞれに工具の入ったケースをさげて，なかなかいそがしそうだった。1日中パタパタと，おびただしい数の切符を送りつづけている機械の軍団が，工具をかかえてあの駅，この駅とめぐる人たちをつくりだすのも面白い。

切符の自動販売機をはじめて使ったときも，スゴイなあと思ったものだ。切符が出てくることより，オツリが出てくることに感心した。それが昭和40年代だったので，いま街にあふれている自動販売機の歴史はそんなに古いものではないと思っていたら，大まちがいだった。

最古の自動販売機は紀元前215年にエジプトの寺院にあったという話を，日高敏氏の『ものの文化誌』で知った。世界最初の自動販売機は聖水を売る装置で，5ドラクマ硬貨(約225円)を入れると，ある分量の水が出てくるものだった。

略図を見ると，大きな水壺のなかに細い支柱が立てられ，その頂上を支点にしたシーソーのようなしかけがつくられている。一方は硬貨を受ける円盤で，もう一方には水路をとざすフタがオモリのように下がっている。壺の上部の口から硬貨を入れると円盤の上に載り，重みでフタが上がって水が流れ出る。水の量の変化で硬貨が落ちると，水も止まる。この装置についての文献を残した紀元1世紀の機械学者で数学者の名前から，「ヘロンの聖水販売機」と呼ばれているそうだ。

それからあとの自販機の歴史には，あらゆる種類のものが登場してくる。鉛筆，タバコ，葉書，菓子，ワイン，ガス，電気。やはりアメリカがいちばん盛んで，ちょう

ど100年くらい前には，自動写真撮影機，新聞自動販売機，自動預金証明装置などが出てきている。光線自動販売装置というのもあった。硬貨を入れると30分だけ照明がともる装置で駅などに置かれ，旅行者が読書したり，手紙を書いたりするのに役立てたそうだ。

1925年のアメリカで，それまではひとつの機械からはひとつの商品しか出てこない単能機だったのが，複数の商品から選ぶことのできる多能機があらわれ，これがいまある自販機のはじまりのようだ。

日本でも，1904年には切手・葉書の自販機を考案した人や，切符の自販機の特許を出願した人もいる。

切符の自販機が正式に国鉄で使われはじめたのは，中山小一郎という人が考案した「中山式入場券自働発売機」で，1929年(昭和４年)のことだそうだ。

自販機のことを考えていたとき，パチンコ台と類型のような気がしてしかたなかったのだが，やはり関連のあることがわかった。この当時，入場券自販機の電動式機械の特許申請もあり，こういう技術がパチンコ台の製作に応用されたと，日高氏の著書に記されていた。

もっと興味深かったのは，そのパチンコ台の技術改革が，こんどは逆に切符の自販機のバージョンアップに影響をあたえたということだった。パチンコ店で人手によって補給していた玉が，「群管理システム」と呼ばれる方式の導入で自動補給できるようになった。それまで切符の自販機もオツリがなくなるたびに詰めかえていたが，これがヒントになって，自動補給のシステムが採用されたということだ。

券売機(切符の自販機)も1968年には金額や日付が印刷される印刷式多能機があらわれるが，まだ印刷インクの制御がうまくいかなかったために，手がよごれるなど評判もよくなかった。その後1971年にはインクも紙も改良されたキレート方式になり，いまではサーマル印刷(熱転写)方式が採用されている。

コイン式ガスメーター

自販機だけの歴史を読んでも，生活のなかに登場してくる「物」や，人々の「考えの豊かさ」がとても身近に感じられる。また，人間の思いつきの相似性なども面白い。

いま「自販機文化」の三大国は，アメリカ，日本，ドイツなのだそうだ。たしかに私たちが自販機で買える商品はふえつづけているが，小銭で買い物をする自販機よりも，すこしまとまったお金で買い物をするカードシステムのほうが興味の対象のようだ。これからは自販機用のカードシステムの開発ももとめられるという。

日高氏のこの著書でアメリカの自販機の歴史のなかに，1887年にあらわれた最初の硬貨投入式ガスメーターの話がある。R・W・ブラウンヒルが特許を得たこの機械は，１回に１ペニー，あるいは１シリングの硬貨で，ある量だけのガスを買うことができたそうだ。

ジェラルド・カーシュの短編ミステリー『詐欺師カルメシン』は，ポンッ！　と音をたててガスコンロの火が消えてしまうところから話がはじまる。「わたし」は自分の部屋で，１ペニー銅貨１枚分のガスがあまりに少ないことに腹をたてて，ガス会社を口汚くののしる。

テーブルに向かいあっていた，胸も体も分厚く大きな老人カルメシンは，「わが若き友よ」とさとす。「平静さ，バランス，それに，客観的にものを考える能力，それがこの世で必要なものなのじゃよ」。

コインのない「わたし」は，１ペニー銅貨の大きさにボール紙を切ってみることを考えるが，その方法はガスメーターをこわすだけでなり，家主にすべてを白状しなければならなくなり，失敗と屈辱の両方を味わうことになるとカルメシンはいう。

優しく年老いた犯罪者，詐欺師の語る壮大な自慢話。ある事情でフランスを端から端まで旅行することになり，途中で流感にかかってしまう。所持品もお金もないままに，とある下宿部屋で厳寒の冬をすごさなければならなくなった。

外は雪，流感の高熱のなかで，ひと晩かかって自称「天才」は案をねりあげると，「あくる朝にはもうガス灯が輝き，ガス暖房も燃えていて，わが輩はもうブルブル震えるのをやめていた。しかもだ，これをわが輩はメーターに１枚のコインを入れもせず，メーターの仕掛をいじくることもせずにやってのけたのさ」。

illustration：Kyoko Takazawa

ガス会社の役人がコインをあつめにやってくる。メーターには消費されたガスの量が示されているのに，コインはない。問いつめられるが，カルメシンは自分は病人だ，なにも知らないという。役人は上司をつれてきて，またたずねる。部屋はあかあかとガスが燃えているのに，カルメシンはシラをきる。

ガス会社からは重役が面会をもとめてきて，必死で事情をたずねる。わけを話してくれれば，罰しないばかりか多額の謝礼を支払うという。そこで彼は「天才の頭脳」で考え出したという，とっておきのタネあかしをして１万フランを手に入れた。

どこまでほんとうかわからない，脚色にみちた話を語るカルメシン自身がテーマでもある。「だがね，１万フランぽっち，何だというんだね。ひよっこのエサ代じゃないか！」と，最後までポーズする。

自動改札機を，おとなが子供用の切符をもって通過したとき，じっさいにどうするのかを営団地下鉄にたずねた。

「混雑しているときは，呼び止めることもできないので，いまJRでも新しい方法を開発しています」。鉄道のハイテクは，やはりJRがリーダーらしい。

出典：日高 敏著『ピラミッドに自動販売機があった・ものの文化誌』(昌文社刊)
中村保男訳：『詐欺師カルメシン』(『ミニ・ミステリ傑作選』創元社)

another
cg world
MOUSE PAD STORY
まっくの助

CGに
王道
なし

CLICK

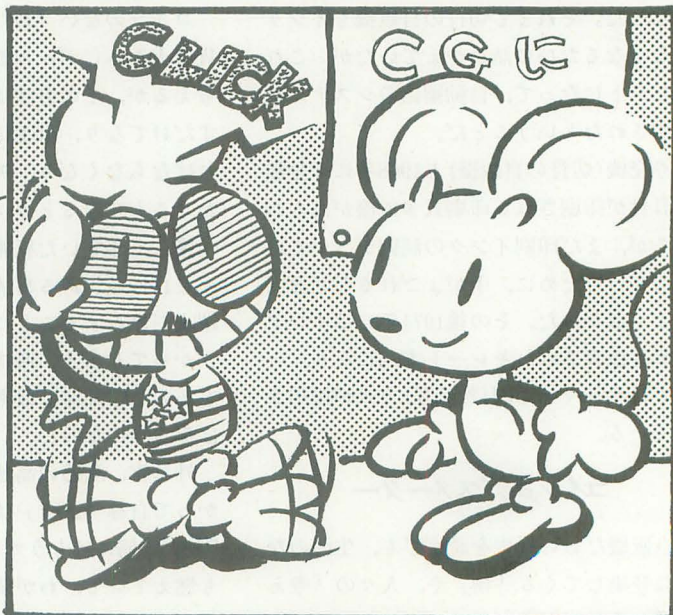
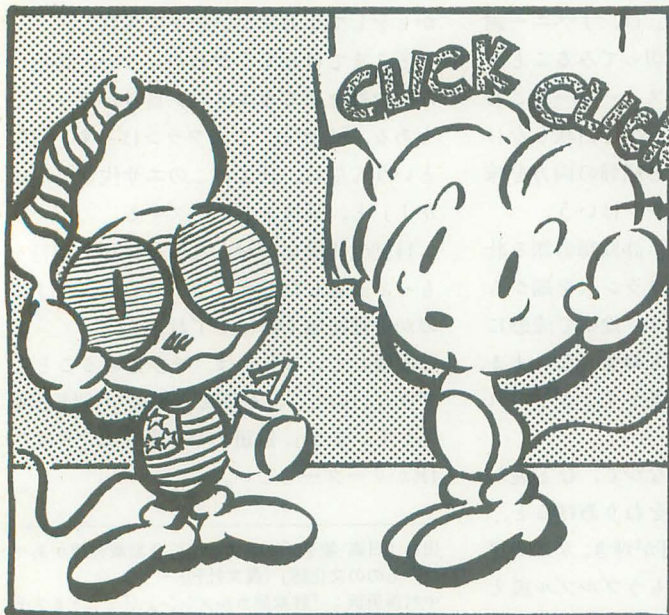
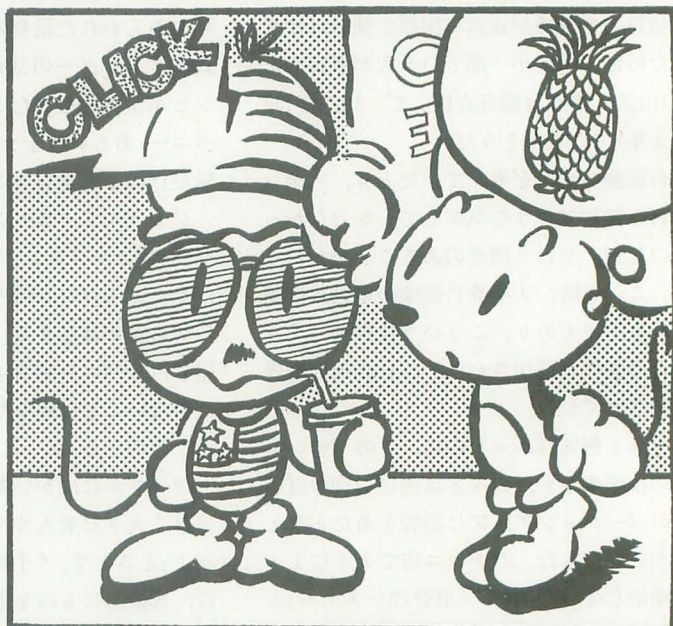
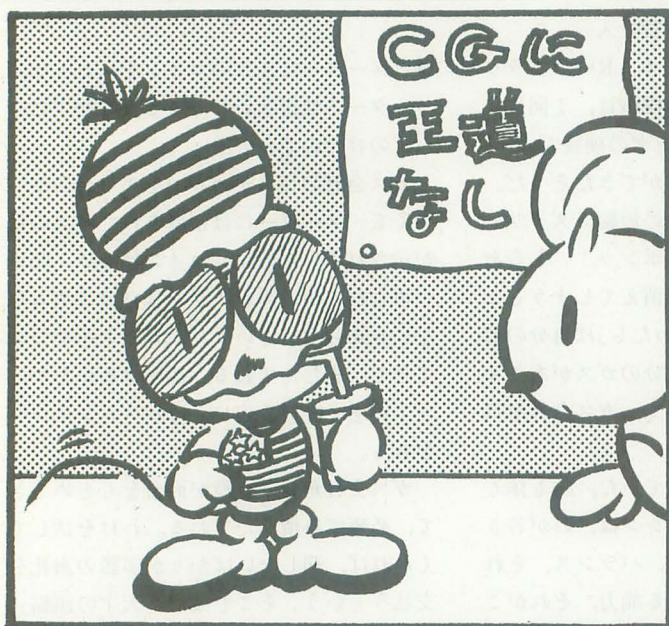
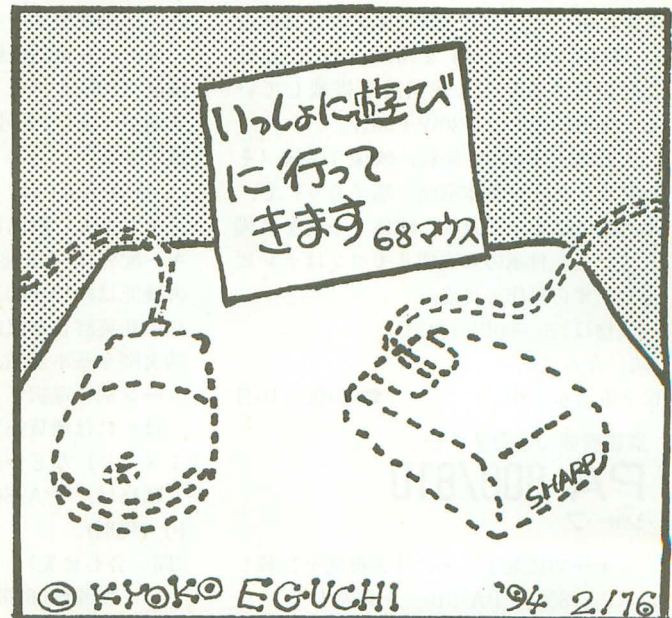
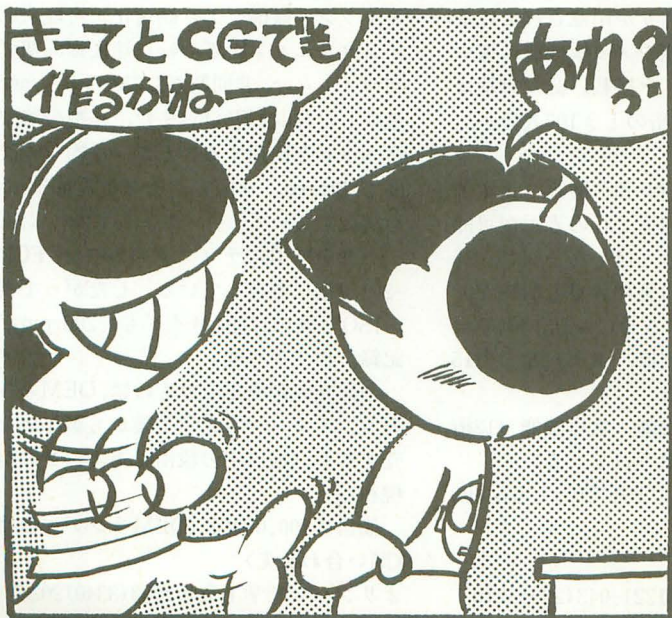
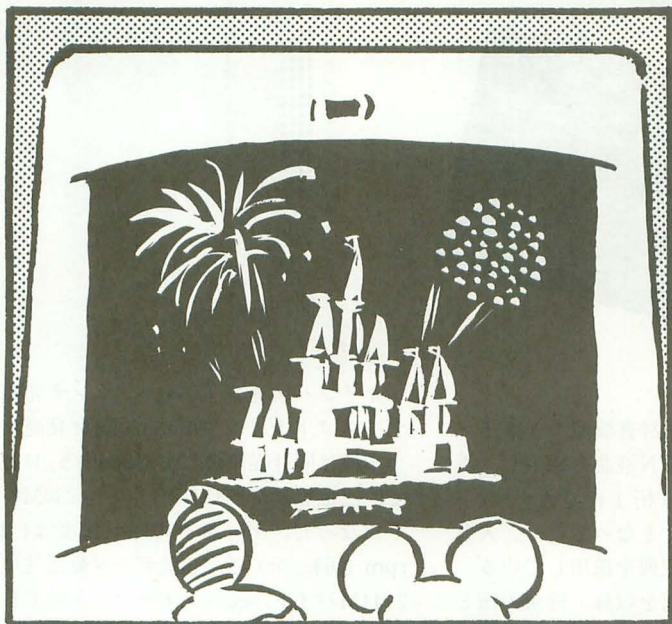
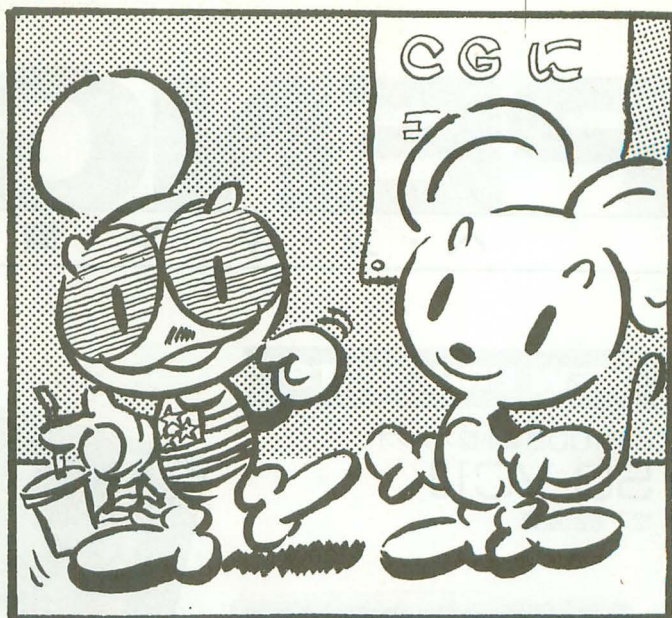
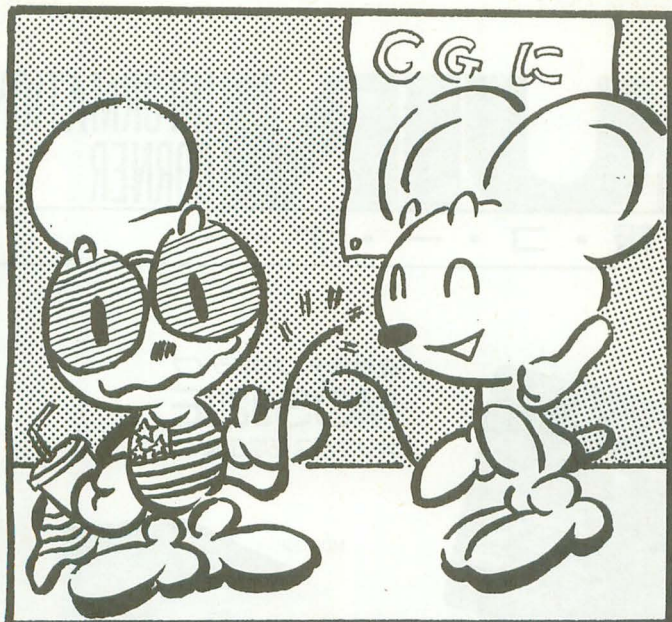
CLICK CLICK

CLICK
CGに

CGに

CGに

After Darkのディズニーコレクションは、
ごっつうかわいいでっせ

さーてとCGでも作るかねー
あれっ?

いっしょに遊びに行ってきます 68マウス
SHARP
© KYOKO EGUCHI '94 2/16

PENGUIN INFORMATION CORNER

ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

ビデオCD搭載小型ステレオ
SC-VC10
松下電器産業

SC-VC10

松下電器産業はビデオCDを搭載した小型ステレオ「SC-VC10」を発売する。

本機はビデオCD（音楽用CDと同じ直径12cmのディスクで74分間の動画と音が再生可能），音楽用CD，静止画の出るCD-G（グラフィクス）の再生が可能。さらに，FMワイド/AMステレオ対応チューナー，カセットデッキを搭載している。ビデオCDの規格は，ビデオデータがMPEG1準拠で，オーディオデータはMPEG1 レイヤー2に準拠している。

スピーカーには，ウーハーの前にパッシブラジエーターという振動体を設けた構造の「D.D.スピーカー」を採用。また，重低音を再生するV･BASS回路を搭載している。実用最大出力が30W＋30W。

入力端子は音声3系統，映像3系統（それぞれ1系統が本体前面に端子あり）で，出力端子は音声1系統，映像2系統を装備している。付属の多機能リモコンはテレビとビデオの操作も可能。

価格は125,000円（税別）。

〈問い合わせ先〉
松下電器産業㈱　☎06(909)1021

辞書機能つき電子メモ
PA-800/810
シャープ

シャープは電子メモに辞書機能を搭載した「PA-800」「PA-810」を発売する。

PA-800

PA-810

「PA-800」は，漢字辞書機能つき電子メモである。画面はFSTN液晶を採用し，漢字を表示する場合は4桁1行でカナ・数字などの場合は12桁1行となっている。入力キーはカナ50音キー配列を採用している。辞書機能は約34,000語を収録。検索方法としては，見出し語引き，音訓引き，部首引き，総画数引きの4通りが用意されている。

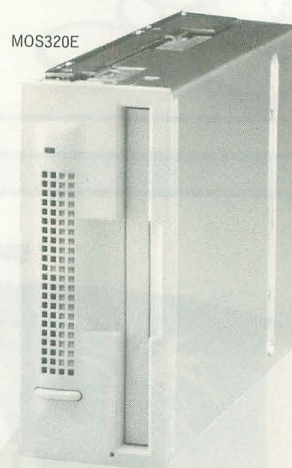
ほかにも電話帳機能やメモ帳機能，電卓機能（10桁1メモリ）を搭載。記憶容量は名前が4文字，番号8桁のとき107人分が記憶できる。

「PA-810」は，英和・和英辞書つき電子メモである。画面は14桁2行の表示が可能。キー配列は「PA-800」とほとんど同じ。辞書機能は約12,000語の英和辞書と約15,800語の和英辞書を収録。さらに，約310例の会話文例を基本会話，空港，ホテルなどの15シーン別に選択できる。

ほかには通貨換算機能，電卓機能（12桁1メモリ）などが搭載されている。

価格は，「PA-800」「PA-810」とも6,600円（税別）。

〈問い合わせ先〉
シャープ㈱☎06(621)1221,043(299)8210

光磁気ディスクドライブ
MOS320E
オリンパス光学工業

MOS320E

オリンパス光学工業は3.5インチ光磁気ディスクドライブ「MOS320E」を発売する。

本機は同社従来機「MOS300E/S」に比べて，記憶容量が128Mバイトから230Mバイトになった。ディスクの回転速度は4,200rpm（回転/分）で，連続データ転送速度は2.44Mバイト/sec（フォーマット時で1.72Mバイト/sec）となっている。さらに，内蔵キャッシュ機構でより高速化が可能。SCSIインタフェイス上での転送速度は同期で5Mバイト/sec，非同期で3.3Mバイト/secを達成した（同社調べ）。また，小型分離型光学ヘッドを採用することで，平均シーク速度で28msec，平均アクセス速度で35msecを実現している。インタフェイスにはSCSI-1・2を採用。ディスクの規格は，ECMA-201規格に準拠している。したがって現行のISO準拠の128Mバイトも4,200rpmでの記録，再生が可能。

今回発売が決定したものは，OEMのみであり，オリンパス光学工業から製品版を発売するかは検討中の段階である（2月24日現在）。

価格は100,000円（OEMサンプル価格）。

〈問い合わせ先〉
オリンパス光学工業㈱☎03(3340)2020

パーソナルワープロ

WD-Y340

シャープ

WD-Y340

シャープはパーソナルワープロ書院「WD-Y340」を発売した。

本機は4書体（明朝体，毛筆体，ポップ調書体，江戸文字）の「書院スーパーアウトラインフォント」をROMで内蔵している。文字の大きさも3～277ポイントまで104種類の設定が可能。約40,000語の国語辞典機能では，単語の意味，用例，類語，対義語の検索ができる。外国の言語についても英・独・仏など10カ国語の欧文書体を搭載している。それに伴う英文スペルチェック機能やワードラップ処理，両端揃え印刷といった機能により，欧文ワープロとしても使用できる。

また，ハガキやカードなどのレイアウト作成も付属ソフトの「はがき屋大将」で対話的に行える。14種類（テープラベル印刷，インデックス印刷など）のアプリケーションを利用することで多彩な印刷が可能になった。ほかにも自動レイアウトや自動罫線，自動文書作成，表計算，グラフ，タイピング練習ソフトなどの機能も専用ソフトの搭載で充実している。本体にはフロッピーディスク（6枚）とACコードが収納できるボックスも用意されている。

価格は140,000円（税別）。

〈問い合わせ先〉

シャープ㈱☎06(621)1221,043(299)8210

ファクシミリ電話機

UX-T1CL

シャープ

UX-T1CL

シャープはファクシミリコードレス電話機「UX-T1CL」を発売した。

本機は，従来機「UX-T1」に留守番電話機能，コードレス電話機能を加えたものである。留守録にはデジタル録音方式（DSP方式）を採用し，巻き戻しなどの操作をせずに録音内容の再生が可能。暗証番号を指定することで個人の伝言を各人別に録音することもできる。コードレス子機は，標準装備1台で3台の子機が増設可能。

また従来機にもあった本体の原稿読み取り部を取り外して使うB4ハンドコピー機能では，ノートなどの綴じた原稿のコピーもできる。またFAX機能では操作のガイダンス機能がある。ほかにも通信終了，用紙切れなどの情報を子機に知らせるインフォメーション機能，FAX送受信指示などを行うリモート機能，子機が使用中でもFAXの送信予約ができるリザーブ送信機能がついている。

価格は98,000円（税別）。

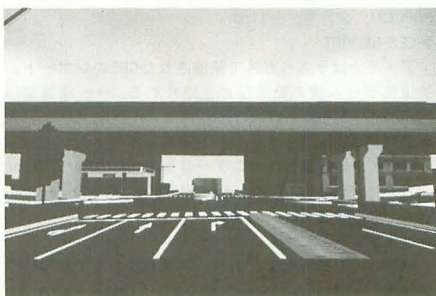
〈問い合わせ先〉

シャープ㈱☎06(621)1221,043(299)8210

INFORMATION

とみんワールド'94

―世界都市博覧会まであと2年

東京都情報連絡室

東京都情報連絡室は都政に対する理解と共感を深めるためのイベントとして，「とみんワールド'94」を開催する。このイベントは1996年に開催される「世界都市博覧会―東京フロンティア」のプレイベントとして行われる。

イベントの内容は「東京の未来」をテーマに，コンピュータグラフィックやバーチャルリアリティ，3Dステレオグラムなどを利用したクイズゲームやCG画展などが用意されている。

開催期間は3月15～31日，場所は東京都庁内都政情報センターと都民広場。

〈問い合わせ先〉

東京都情報連絡室事業課　☎03(5388)2265

国際交流イベント

パーソナルコンピュータの未来像

日本パーソナルコンピュータソフトウェア協会

日本パーソナルコンピュータソフトウェア協会は国際交流イベント「パーソナルコンピュータの未来像」を4月20日に開催する。

同イベントはパーソナルコンピュータのルーツとその未来像というテーマのもとに，アラン・ケイ氏を招き21世紀に向けたパーソナルコンピュータの理想像を講演する。ほかにもダグラス・エンゲルバート氏，ポール・サフォー氏などの講演やパネルディスカッションが予定されている。

場所は東京都千駄ヶ谷の日本青年館大ホール。時間はPM1：00～PM7：00。入場料は前売で5,400～9,000円，当日で6,000～10,000円。

〈問い合わせ先〉

㈳日本パーソナルコンピュータソフトウェア協会　☎03(3253)1098

オリジナルマウスパッドプレゼント

ツクモ電機

ツクモ電機は1993年10月にオープンした「DOS/Vパソコン館」の好評に応えて同社のCMキャラクタとして起用されている越智静香をデザインしたオリジナルマウスパッド（定価1,500円）を5名の方にプレゼントする。ハガキに住所，氏名，年齢，職業を明記のうえ，下記あて先に送ること。当選者の発表は商品の発送をもって行う。締め切りは4月18日到着分まで。

〈問い合わせ先〉

〒101　千代田区外神田3-2-14

ツクモ電機・広告部Oh!X係

FILES

Oh!X

このインデックスは，タイトル，注記――著者名，誌名，月号，ページで構成されています。これから新しい生活が始まる人も特に変わらない人も季節の変わり目になにか新しいことに挑戦してみてはいかがですか。

参考文献
I/O　工学社
ASCII　アスキー
コンプティーク　角川書店
C Magazine　ソフトバンク
テクノポリス　徳間書店
電撃王　主婦の友社
PIXEL　図形処理情報センター
POPCOM　小学館
マイコン*BASIC Magazine*　電波新聞社
My Computer Magazine　電波新聞社
LOGIN　アスキー

一般

▶NEWS&SOFT Radar!
ブラザー工業の「名作文庫ソフト」シリーズや，魔法株式会社によるX68000版「龍虎の拳」「餓狼伝説SPECIAL」発売のニュースなど。――編集部，LOGIN，4号，6-7pp.

▶これであなたも業界人
雑誌やヒットゲームを作ったりするゲーム業界の人たち。業界人の生活スタイルやメリット，業界人になる方法を紹介する。――編集部，LOGIN，4号，91-103pp.

▶THE NEWS FILE
セガの「サターン」のスペック，三洋電機の3DOマシンなどゲーム機のニュースから，東京オートサロン'94の模様まで。――編集部，LOGIN，4号，104-111pp.

▶SFロボットを作っちゃおう！
若手ロボット研究者たちが，現代の技術を駆使してSF映画やアニメ作品のロボットたちを検証している。――鹿野司，LOGIN，4号，162-165pp.

▶特捜情報最前線
今月のTOPSALESでは「餓狼伝説2」が第4位を確保。コミケット45で見かけたコスプレ特集，アーケードの新作紹介など。――編集部，コンプティーク，3月号，17-28pp.

▶安田均のゲームさいころジー
CES体験記。会場の模様や展示の内容だけでなく，ラスベガスのホテル事情などもあわせてレポートする。――安田均，コンプティーク，3月号，94-96pp.

▶NEWSCOLLECTORS
WINTER CESスペシャルと題して，米国CESの話題を中心にトピックを伝える。3DO社長の記者会見，他社の次世代機の動向など。――編集部，電撃王，3月号，8-15pp.

▶「バーチャファイター」の魅力と可能性
キャラクター別攻略法，CG技術についてのインタビューなどで構成された「バーチャファイター」総力特集。――電撃王，3月号，30-42pp.

▶GAME MUSIC専科
ラジオDJ国府田マリ子と葉山宏治の対談ほか，B-univ初ライブのレポート，CDやVIDEOの新作情報など。――編集部，電撃王，3月号，131-133pp.

▶輝け!!POPCOM OSCAR
POPCOM創刊から現在までのあいだに発売されたゲームから，最優秀作品および各部門賞を設定し，読者代表のアンケート集計をもとに授与する。――編集部，POPCOM，3月号，5-11pp.

▶CES REPORT
アメリカはラスベガスで開催されたCESのレポート。次世代ゲーム機の動きなどを紹介する。――編集部，POPCOM，3月号，111-114pp.

▶最新マルチメディア・マシン試用レポート
CD-ROMを内蔵したモニター一体型のIBMのPS/V Visionの性能をチェックする。――編集部，マイコンBASIC Magazine，3月号，35-39pp.

▶映像世界を広げるニューハード
特殊なサングラス上にカラー映像を映し出す世界初の携帯型バーチャルビジョンテレビ，「Virtual Vision Sport200」を取り上げレポートする。――編集部，マイコンBASIC Magazine，3月号，40-41pp.

▶先生と生徒のためのBASICプログラミング講座
今月のテーマは「いろいろな数字に強くなろう！」。年利・利息額や歩留まり率をBASICを使って計算してみよう。――東幸太，マイコンBASIC Magazine，3月号，66-69pp.

▶速報！「サターン」
今秋発売予定のセガのゲーム機「サターン」のスペックや画面写真を紹介。――編集部，マイコンBASIC Magazine，3月号，138p.

▶山下章のウインターCES'94レポート
ラスベガスで行われたCESの模様を伝える。次世代コンシューマー機の動向，パソコンのソフト最新作の紹介，ハイテク界の名品，珍品コレクションなど。――山下章，マイコンBASIC Magazine，3月号，139-151pp.

▶NEW PRODUCTS
オリンパスの光磁気ディスク「MOS300S01A」や，高速FAXモデム「通信ポコMP1414F」など，最新のコンピュータ周辺機器の情報を紹介する。――編集部，I/O，3月号，18-21pp.

▶特集　マルチメディア
マルチメディアの全体像とその作り方について解説する。――大澤文孝ほか，I/O，3月号，32-45pp.

▶マルチメディアの行方
電子ブックの試用レポートとこれからの課題について考える。――奥野雅之，I/O，3月号，119-122pp.

▶特集II　コンピュータサウンド
主としてWINDOWS用の音楽事情を紹介するが，MIDI音源やキーボードも多数登場。――編集部，ASCII，3月号，249-272pp.

▶高集積度特集・次世代ゲームマシン
松下電器の「REAL」，ATARIの「JAGUAR」，COMMODOREの「AMIGA CD32」などを取り上げて批評する。――編集部，ASCII，3月号，290-296pp.

▶PRODUCTS SHOWCASE
A3対応ページプリンタ2機種，各種パソコン用スピーカシステム，松下電器の新型平面ディスプレイ「フラットビジョン」などを紹介。――編集部，ASCII，3月号，297-312pp.

▶DIGITAL WATCH
松下電器の「REAL」を取り上げ，日米のアーティストが対談。CD-ROMというメディアとゲームとアートの関係に話は及ぶ。――桝山寛＋David D'Heilly，ASCII，3月号，360-363pp.

▶バカパパのモノを買い物
今回はハコなどの収納系を集めている。――バカパパ，ASCII，3月号，364-365pp.

▶未来派パソコン通信の研究<5>
ソニーのSMDシリーズを取り上げて，今後のパソコン通信を考える。――原田洋平，My Computer Magazine，3月号，151-153pp.

▶ビジネスマンのための情報管理術
前回に続いて「ザウルス」の活用レポート。――塚田洋一，My Computer Magazine，3月号，156-160pp.

▶HARDWARE FORUM
各社のモニタを取り上げてその性能や特長を紹介する。――編集部，LOGIN，5号，152-155pp.

▶アメリカ最新事情
今回はCESに出展された最新ソフト情報と，「Mac World Expo」の模様を伝える。――編集部，LOGIN，5号，174-181pp.

▶架想楽園へ行こうVer.2.01
VR研究の最前線をレポートし，人間の五感をコンピュータがどこまで気持ちよくしてくれるかを探る。――中田宏之，LOGIN，5号，192-197pp.

▶絶滅メディアに学ぶ
外部記憶装置の歴史を振り返る。幻の3インチFDDも登場。――編集部，LOGIN，5号，198-201pp.

X1/turbo/Z

X1シリーズ

▶FINAL DEFENSE LINE
縦スクロールのアクション・シューティング。――宮下智基，マイコンBASIC Magazine，3月号，114-115pp.

X68000

▶X68新聞
X68000ソフト紹介のページ。「ジオグラフシール」と「マッドストーカーX68」を取り上げる。――編集部，LOGIN，4号，124-125pp.

▶SUPER SOFT EXPRESS
ゲームの情報コーナー。X68000は「ジオグラフシール」「マッドストーカーX68」の紹介ほか，「魔法大作戦」発売決定の情報など。――編集部，コンプティーク，3月号，37，47-49pp.

▶NEW GAME REPO!!
各機種用の最新ソフトを開発中のものまで紹介する。X68000用は「マッドストーカーX68」「ジオグラフシール」「あすか120%」など。――編集部，テクノポリス，3月

号, 39, 42, 48pp.
▶DO-JIN SOFT FAN!!
今月はコミケット45の特集。同人ソフトを一覧表の形で公開。X68000用は「PARTIAL AXIOM」など。──編集部, テクノポリス, 3月号, 67-75pp.
▶今月の電撃王
電波新聞社の「エキサイティングアワー/出世大相撲」など, 2月の話題作をピックアップ。──編集部, 電撃王, 3月号, 21p.
▶新作王
ポリゴンを使ったX68000用3Dシューティング「ジオグラフシール」など, ゲームの最新作を紹介するコーナー。──編集部, 電撃王, 3月号, 163p.
▶Hot Press
エグザクトの新作「ジオグラフシール」, 往年の名作ソフトを安価に供給する「名作文庫ソフト」シリーズなど, 各機種向け最新ソフトを紹介するページ。──編集部, POPCOM, 3月号, 19, 21pp.
▶HOT PRESS+1
話題の新作21本を厳選して紹介する。X68000用は「スーパーリアル麻雀PIV」「エキサイティングアワー/出世大相撲」などが登場。──編集部, POPCOM, 3月号, 24, 25pp.
▶「ドラゴンナイト4」のすべて
3月にX68000版の登場が噂されている「ドラゴンナイト4」。前作の復習もからめながら, その内容とグラフィックの一部を紹介。──編集部, POPCOM, 3月号, 104-107pp.
▶New PRODUCTS
マイクロウェアシステムズから発売される「ビデオPC for X68000」をレポートする。CDに74分間の動画と音声が記録できるという「ビデオCD」規格。ビデオCDと再生ボードを, 仕組みも含めて解説する。──編集部, マイコンBASIC Magazine, 3月号, 42-43pp.
▶PUNCH!
2人でも遊べるボクシングゲーム。ストレートとアッパーを繰り出し相手をリングに沈めるのだ。──中山聡, マイコンBASIC Magazine, 3月号, 116-117pp.
▶スシ喰いねぇ!
客が注文した寿司を, 一定時間内に届けよう。レベルが上がると客の注文が殺到するぞ。──濱口和彦, マイコンBASIC Magazine, 3月号, 118-120pp.
▶イースIV〜THEME OF ADORU 1993〜
PCエンジンで発売された「イースIV」からのミュージックプログラム。──重長孝之, マイコンBASIC Magazine, 3月号, 127-129pp.
▶SUPER SOFT HOT INFORMATION
各機種の新作情報。X68000用は「ジオグラフシール」「宝魔ハンターライム」「B-Field!」など。──編集部, マイコンBASIC Magazine, 3月号, とじ込み付録9p.
▶AV STRASSE
「OS-9/X68030 X Windows V11.5」を紹介。──編集部, ASCII, 3月号, 333-336pp.
▶ON-LINE SOFTWARE INDEX
大手主要ネットにアップロードされたソフトを紹介。X68000用にはリソースエディタ「MenuDesign.x」など。──編集部, ASCII, 3月号, 401-407pp.
▶CD-ROMソフトをX68で楽しむ
計測技研のCD-ROMソフト3本を紹介。──高橋雄一, My Computer Magazine, 3月号, 60-61pp.
▶なんでもQ&A
今回はOSのパイプ機能とリダイレクト機能について答える。──編集部, My Computer Magazine, 3月号, 174-175pp.
▶NEWS&SOFT Radar!
「スーパーリアル麻雀PIV」発売決定のニュースなど。最新版ゲームリリース情報も掲載。──編集部, LOGIN, 5号, 6-9pp.
▶NEW SOFT
エグザクトの最新作「ジオグラフシール」をはじめ, 各機種用の新作ゲームを紹介する。──編集部, LOGIN, 5号, 22p.
▶X68新聞
電波新聞社の「エキサイティングアワー/出世大相撲」の紹介と「龍虎の拳」「餓狼伝説SPECIAL」移植決定のニュース。──編集部, LOGIN, 5号, 160-161pp.
▶GAME BUSTERS!
ファミリーソフトの格闘アクション「マッドストーカー X68」など各種ゲームソフトの攻略法を伝授する。──編集部, LOGIN, 5号, 238, 239pp.
▶簡単で本格的な2次元画像作画教室 2
「MATIER」のメッシュ変形を使ってシュールリアリズムイラストに挑戦する。スキャナの取り込みから, 変形, 合成までを解説。──長谷川一光, PIXEL, 3月号, 115-119pp.
▶SX-WINDOWプログラミング<第5回>
今回は, スケルトンを改造してSX-WINDOWの負荷の状態を計測するパフォーマンスメーターを作成する。アイドルイベントの発生回数をカウントして負荷を測るソフトだ。──吉野智興, C Magazine, 3月号, 138-141pp.

ポケコン

PC-E500
▶AVOIDING THE DOTS
ワンキー・アクションゲーム。上昇下降を操作して, ドットをよけつつ画面の右端を目指す。──西野陽一, マイコンBASIC Magazine, 3月号, 121p.

新刊書案内

イラストで読む
パソコン入門
ロン・ホワイト著
福崎俊博訳
インプレス刊
☎03(5269)7131
B4変形判 147ページ
2,800円(税込)

最初の行からかましちゃうけど, 「この本を買え」。タイトルだけみると, そこらに転がっている「まんがでわかるパソコン入門」みたいだけど, 中身のレベルは段違い。さすが, アメリカ。日本で出版されるパソコン入門書は, 「中身は知らなくてもいいから, 使い方を覚えろ」的な「型から入れ」式教習本がほとんどだった。これでは, すぐにでもパソコンを使わねばならないビジネスマンにはよくても, コンピュータという存在に向き合おうとする本誌読者には生ぬるい。しかし「イラストで読むパソコン入門」(原題は「How Computers Work」)はまったく違う。とにかく, ハードウェアからOSまで, ブートアップからネットワークまで, パソコンを解剖してみせ, 原理をわかりやすく解説し, コンピュータの動作概念を叩き込んでくれるのだ。「これは〜という名前で〜に使うものです」ではなく, 「これは〜という仕組みで〜なときに〜ですから, 〜の仕事をするのです」という流れだ。具体的な規格名や人前にひけらかす知識はくれないが, 基本原理はしっかり理解できるから, 新しい技術がやってきても応用が利く。アメリカの本だから, IBM PC=パソコンとして解説しているけれども(ちなみに, Macintosh版もある), インテルの石+DOSというマシンならそのまま応用が利くし, X68000でも基本原理は一緒だ。386+OSのあたりを68000+Human68kにうまく置き換えて考えれば, なんとかなる。心配することはないだろう。

イラストはドローイングソフト(おそらくは, PostScriptソフト)で描かれたもの。まず文章で基礎知識が解説され, 次に, 見開きいっぱいを使ったイラストとそのコメントで動作が解説される。イラストレーターとライターの二人三脚による傑作だ。高いけど, 買って損はない。ただ, コンピュータ自体に興味のない実用オンリーユーザーには無意味な本。 (K)

情報喪失の時代
ビル・マッキベン著
高橋早苗訳
河出書房新社刊
☎03(3404)1201
四六判 263ページ
2,500円(税込)

現在, 自然破壊に関する情報はテレビのドキュメント番組で数多く見られる。そういった情報が与えられれば, その状況を改善したいと思うのが当然だろう。ところがその流れはあくまでも一部である。著者はその原因がテレビの与える情報にあるのではないかと考えた。そこで2通りの1日を実験的に過ごした。ひとつは, ある1日は放送されたケーブルテレビ局の番組のすべてを録画し, それを見る。もうひとつは, 池のある山の頂上でキャンプをした1日。本書はこの2日間を対比しながら, テレビが人類と自然の関係にどのような影響を及ぼしたか考察している。

左利きは危険がいっぱい
スタンレー・コレン著
石山鈴子訳
文藝春秋刊
☎03(3265)1211
四六判 398ページ
2,000円(税込)

昔から天才は左利きが多いといわれてきた。一例を挙げるとレオナルド・ダ・ビンチ, アインシュタイン, モーツァルトなどがそうである。また, 活躍する人の多い一方, さまざまな病気を抱えるグループのなかにも左利きの人の割合が異常に多いともいわれてきた。ほかにも日常習慣のなかで, 左利きが偏見や差別の対象になったり, 右は善で左は悪といった発想までがあった。

本書は左利きの存在を遺伝的, 医学的, 心理学的な側面はもちろん言語, 文化, 技術, それに社会環境に至るまで考察し, 左利きの謎に迫る。左利きの人も, そうでない人も楽しめる一冊である。

3月号の付録ディスクはあまり使えるものがありませんでした。私はMOは持っていませんし，Z'sSTAFFもMATIERも持っていませんし，SX-WINDOWはver.2.0ですし，メモリは2Mバイトなのでほとんどのプログラムが使えません。もう2Mバイトではやっていけないものなのでしょうか。

栃木県　高崎　要太郎

SX-WINDOW用のプログラムを使いたいのならSX-WINDOW ver.3.0を導入するのは当然だと思うので，SX-BASICなどを使うなら，なにはともあれver.3.0を入手してください。ツールの充実度はもちろん，動作速度もかなり違います。そうなるとメモリも増設するのが当然なのですが……。

では本題に入りましょう，メモリ2MバイトだとSX-WINDOW ver.3.0がまったく使えないか？　というと，そうでもありません。SX-WINDOWの機能はかなり限定されますが，いくつかのアプリケーションを起動することはできます。

まず，CONFIG.SYSをできるだけ軽くします。たとえば図1のような構成です。

次にSX-WINDOWを立ち上げ，スタートアップメンテ.Xを起動して必要のなさそうなものをはずします。すべてのものをはずしてもかまいません。IFM.XとIVM.Xは絶対に登録してはいけません。

さらにアイコンメンテ.Xでパターン一覧を呼び出し，必要のなさそうなアイコンを片っ端から削除していきます。

ひととおりの作業を終えたら出ているウィンドウを閉じて再起動してください。

こういった作業を行うとメインメモリ2Mバイトの状態でもSX-WINDOW起動時に約1.2Mバイトのフリーエリアが確保されます。

この状態でならSX-BASIC，ウィンドウデザイナ，ウィンドウエンジン，おまけにシャーペン.Xと，作業に必要ないくつかのディレクトリウィンドウを同時に開くことができます（ここでは6つ開いていた）。

図1　CONFIG.SYSの例

```
FILES       = 93
BUFFERS     = 20
VERIFY      = on
BREAK       = On
DIRSCH      = on
DEVICE      = ¥SYS¥FLOAT2.X
DEVICE      = ¥SYS¥FSX.X
SHELL       = ¥SHELL¥SXWIN.X -s -al
```

それぞれのツールにはSX-BASICのプログラムがロードされています。ここで試したのは今月号のグラフィックエディタ（それなりに大きな配列をとる）ですが，動作には問題ありませんでした。この時点での残りメモリは150Kバイト程度です。

ただし，システムアイコンは黄色になるので，ウィンドウデザイナを起動するときはSX-BASICで実行中のウィンドウ（ウィンドウエンジン）を閉じるなどとしたほうがいいかもしれません。

これでSX-BASICの雰囲気を味わうことはできますが，作業を続けているとシステムからうるさくウィンドウを閉じろといわれるはずです。さらにこの状態では65536色のグラフィックが使用できません。まあ，このあたりを使うと12Mバイト実装していてもすぐにメモリが足りなくなるので，それほど深刻な問題ではありませんが。もともと無理なことをしているのですから，嫌なら素直にメモリを増設しましょう。

次にZ's-EX&MATIER-EXです。

これらはメモリ食いの代表なのですが，グラフィックツールから呼び出すのではなく，拡張部分だけを取り出して単独で使用することもできなくはありません（あえてできるとはいわない）。

ここではZ'sSTAFFもMATIERも持ってなくてメモリも2Mバイトしかないという人向けに解説します。それぞれの環境によって操作法が違いますし，間違えるといとも簡単に暴走してしまいます。各自の責任において実行するようにしてください。

ではまず，起動システムを整理します。RAMディスクを登録してはいけません。ASK68KとかHISTORYとかははずしておきます。

次にMAT_EX.XのあるディレクトリにCOMMAND.XをMAT.Xという名前でコピーしておきます。いきなり眉をひそめる人もいるでしょうが気にせず続けます。

続いて，EFFECTディレクトリの内容をパスの通ったところに入れておいてください。なお，付録ディスクのMAT_EX.SYSには誤りがあるのでリストのように修正しておいてください。これで下準備は半分終わりです。

とりあえず，

　A>MAT_EX

でMAT_EX.Xを起動します。するとコマンドシェルが立ち上がるので，この状態で，

　A>PROCESS

を実行してみてください。メモリの使用状況が表示されますので，最後のPROCESS.Xの先頭アドレスと最終アドレスをメモしてEXITします。これをもとにエディタでMAT.$$$というテキストファイルを作成します。

ファイルは3行です。1行目は空行でかまいません。2行目と3行目が核心です。

リスト1　MAT_EX.SYSの変更

```
512                                ← 必ず指定すること
:バージョン                            ← この行も変更
        0, 0
        VERNO.X
:PICFILER
        0, 0
        picfiler.x
:CUTFILER
        0, 0
        cutfiler.x
:ALTERNATE
        0, 0
        alternate.x
:PERSPECTIVE
        0, 0
        perspect.x
:MASK
{
        :MASK
                0, 0
                mask
        :MASK PAINT
                0, 1: 0-1, 0
                maskpaint.x
        :緑→マスク
                0, 0
                GtoMask
        :Magic Wand
                0, 2: 0-9, 4: 0-9, 4
                WAND
}
:EFFECT
{                       ← 階層 { } でくくる
        :twirl
                0, 1: 0-9, 5
                twirl.x
        :うにょうにょ
                0, 1: 0-9, 5
                twirl2.x
        :収差
                0,1 : 0-9,5
                shusa.x
        :MONOTONE
                1, 0
                MONOTONE.X        /G

                :                   ( 中略 )

:SAMPLE
{
        :AtoG
                1, 0
                atog
        :AtoG2
                1, 0
                atog2
        :Reverse Mask
                0, 0
                REVMASK
        :Sound Filer
                0, 0
                soundfiler
        :Triple
                0, 0
                triple
        :Resize
                0, 0
                resize
        :Lupe
                0, 0
                lupe       ← ここの綴りを直す
        :EXECUTE
                0, 0
                exec
}
```

先ほどメモした値が$100000を超えていなければ,

U0 100000
U1 180000

のようにしてください。これで準備は終わりました。

いったんEXITしてから,再度MAT_EXを起動して,

A>SCREEN 1 3 1

を実行し,

A>EXOPEN

と打ち込んでください。無事にMATIER-EXが立ち上がったはずです。ここでマウスカーソルしか見えないときは画面の初期化がうまくいっていません。リセットしてやり直してください。

さて,先ほど作ったMAT.$$$について解説しましょう。これはMATIERが使用しているワークエリアの情報を記載したものです。このファイルが残っていればMATIER使用中にうっかりリセットしたり,暴走したりしたときでも裏画面を含めて描画中のデータを復活することができます。

本来1行目には作業日時が入ります。2行目のU0にはアンドゥ画面のアドレス,3行目のU1には裏画面1のアドレスとなっています。環境の違う人は以下のように変更してください。

先ほどのような状態でPROCESS.Xを実行したときにPROCESS.Xの確保したアドレスに着目します。このときの最終アドレスから80000_Hを引いた値をU1に,さらに80000_H引いた値をU0に設定すればよいでしょう。もちろん,U0がPROCESS.Xの先頭アドレスより小さくなってはいけません。

これでMATIER-EXが使えます。ひととおりの機能が使えますが,アドレスを間違うと非常に危険です。万一,裏画面にゴミが出るようならデータと環境を再確認してください。

これらの対応は各ツールを「とりあえず動かす」ためのものです。作業環境としては快適とはいえませんし,無理やり動かしているので動作保証もしかねます。やはり,できるだけメモリを増設されることをおすすめします。

付録ディスクに収録されていたMorph!でPICファイルが読み込めません。どうしたらいいのでしょうか。　愛知県　堀内　薫

まず,PICファイルの大きさを確認してください。Morph!では256×256ドット65536色モードの画像以外は受け付けません。通常の512×512ドット画像はグラフィックツールなどで一度縮小するか,一部分だけを切り出して使用してください。

画面に表示されている絵の真ん中をPIC.Rを使って切り出す場合,

A>PIC -S128,128,383,383 TEST

のような手順になります。

また,ここでいうPIC画像はいわゆるAPICであり,DōGA PICではありませんので注意してください。

Z-MUSICver.2.0ではデータの場所などを設定する環境変数が変わっているようですけど,従来のままではなにか支障があるのでしょうか。　石川県　山口　尚

環境変数zmusicにはAD PCMデータの場所やZPDデータの場所など,Z-MUSICで使用するファイルパスのほとんどが指定されています。Z-MUSICはカレントディレクトリに指定されたファイルがみつからなかった場合,環境変数zmusicに指定されたパスを順番に探していきます。

この便利な環境変数も,システムのコマンドライン制限のため,256文字以上は使えません。Z-MUSICで使うようなたくさんのディレクトリ構成をフルパスで指定するとすぐにあふれてしまいます。特にver.2.0ではPCMファイルの拡張で設定するデータも増え,切実な問題になっていました。

そこで導入されたのが新しい環境変数zmusic0, zmusic1などです。

これは,従来のzmusicを解釈するときにさらに参照されるものです。基本的には,

zmusic=/0B:¥DATA¥
zmusic0=ADPCM;RHYTHM

であった場合なら,

B:¥DATA¥ADPCM
B:¥DATA¥RHYTHM

という2つのディレクトリがファイルサーチの対象になります。

この機能はZ-MUSICver.2.0のものですが,1993年10月号の付録ディスク「秋祭りPRO-68K」に収録していたバージョンでも同様に動作します。

ADPCM_LIST=TEST.CNF

などのようにしている場合は,データによってはZMUSIC.Xの起動オプション指定(-W)でワークエリアを拡大しないとADPCM音が鳴らない場合があります。注意してください。

0～9まで,いくつかの環境変数に分けて指定できますので,最大256文字×11の環境変数が扱えます。これだけあればデータを深い階層ディレクトリの底に入れておきたいという人でも安心です。

ただし,あまりに長い環境変数を使用すると環境のためのメモリが足りなくなることもありますので,必要ならばCOMMAND.Xの起動オプションで環境エリアを広げるようにしてください。たとえば,

COMMAND.X -E:10

のようにすれば3072文字分のエリア(256×10+512)が確保されます。

そのほか,もともとZ-MUSICシステムは,データさえあればディレクトリ構成には依存しないようになっていますので,あえて,ついてきたディレクトリ名や構成にこだわる必要はありません。極論すれば,すべてのAD PCMファイルをひとつのディレクトリに入れてしまえば環境変数で悩むこともなくなります。

最後になりましたが,環境変数があふれさえしなければ,従来の指定方法のままでもなんら問題はありません。

(中野　修一)

質問にお答えします

日ごろ疑問に思っていること,どんなことでも結構です。どんどんお便りください。難問,奇問,編集室が総力を挙げてお答えいたします。ただし,お寄せいただいているものの中には,マニュアルを読めばすぐに解答が得られるようなものも多々あります。最低限,マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名,システム構成,必要なら図も入れてこと細かに書いてください。また,返信用切手同封の質問をよく受けますが,原則として,質問には本誌上でお答えすることになっていますのでご了承ください。なお,質問の内容について,直接問い合わせることもありますので電話番号も明記してください。

宛先:〒103　東京都中央区日本橋浜町
3-42-3
ソフトバンク株式会社出版部
Oh!X編集部「Oh!X質問箱」係

FROM READERS TO THE EDITOR

すっかり風の薫る季節になりました。ビルの合間から吹きつける風も心地よく感じます。会社の帰り道，近くの公園にふらりと寄ってみる。目的は，もちろん宴会ですね。花びら舞う桜の樹の下で飲めや，唄えやドンチャン騒ぎ（冗談）。

◆2月号の特集で再度X-BASICを見直しました。N88-BASICではプログラムを組んだことがありますがX-BASICではありませんでした。N88-BASICと比較すると意外と（失礼しました）使いやすく，スピードは決して速くありませんがそこそこ使えそうに思います。もう一度この機会にX-BASICでプログラムを組んでみようかなと考えています。また，このような特集をお願いします。　壁谷　善嗣(34)愛知県

◆フラクタルがたったあれだけのリストでできてしまうということに驚きました。すごいですね。私は記事を見て電車のなかにもかかわらず「うおー！」と声を上げてしまいました。　砂原　弘幸(22)千葉県

◆2月号の特集のおかげで久しぶりにX-BASICを起動してみました。リストを打ち込んでいると指が思うように動かなくなっていることに気がつきました。やはりリストの打ち込みも日頃の鍛錬が必要だなと痛感しました。　加藤　昌和(34)新潟県

X-BASICの特集ということで，多くの方にX68000に触っていただけたかなと思います。パワーユーザーの方には物足りなかったでしょうが，ハガキを見ると新規ユーザーの方もけっこういるようなので，ゲームだけでなくいろんなことを楽しんでみてくださいね。

◆駅の発車メロディとピコー・ソングを早く聴きたい。でも，SC-55をもっていない。ちなみに僕は渋谷駅のメロディ（ほかでも使っていると思う）が好きです。また，地下鉄半蔵門線の表参道駅にたまに流れる低い発車ブザーの音も好きです。進行方向が逆のホームでは違う音です。　玉木　俊秀(25)神奈川県

◆やられた！　巣鴨駅/新宿駅の発車メロディ。私はそのうち京阪電車の始発メロディと新幹線の車内オルゴールを作って送ろうと思ってたのに……。もう二番煎じになるからやらないけど，それにしても悔しいぞ！　山下　智也(23)大阪府

このメロディには，「やられた」といった方が多かったようです。最近は街のいろんなところで音楽が使われています。皆さんもアイデア勝負でがんばってください。

◆ワンチップIC工作入門が面白かった。ちょうどM³ロボットを買ったときだったので今度改造してみようと思っている。　直井　崇仁(23)神奈川県

図の部分でわかりにくいところがありました。どうもすみませんでした。今月号の「ごめんなさいのコーナー」を見てください。

◆NHKでコンピュータ教育の特集をしていたので見ていた。僕の住んでいる街は小中学校のコンピュータ保有率が100%なんだそうだ。それはそうとコンピュータ教育を進めているいちばんの理由が景気対策だと知ったときは，なんだか幻滅してしまった。　岡元　訓(18)神奈川県

そんなことで本当に子供のためになるのでしょうか……ちょっと心配。

◆X-BASICが進化してSX-BASICとなりSX-WINDOWのプログラミングが簡単になるといいなあ。　広井　誠(32)新潟県

進化したかどうかは怪しいですがSX-BASIC（暫定版）を3月号の付録ディスクで発表しましたが使ってみましたか？

◆長年つれそったX68000ACEを養子に出しました。SASIのHDドライブが行方不明になって以来隠居の身であったパソコンに第2の働き場を与えてくださった先輩，ありがとうございます。　鈴木　貴久(19)神奈川県

X68000ACEの第2の人生（？）はどんなんだろう。そんなことやあんなこと，危険にさらされてなければいいんですが……。先輩，大切にしてあげてくださいね。

◆中国へ行ってきました（広島や岡山じゃないぞ）。タクシーに乗って中国語でホテルの名前を告げたのですが通じません。もちろん日本語は無理で，困り果てて英語で告げると通じました。なんじゃらほい。　末吉　克行(25)兵庫県

このあいだ中国のタクシーの数が東京よりも多くなったという話を聞いたが，それは中国全土と比べたのだろうか？　そのときテレビに写っていた都市は北京だったのですが……。

◆近くのレストランが“リッチドレスデン”から最近“リッチ”がなくなり，ただの“ドレスデン”になりました。不況は厳しいようです。　渡場　雄(19)青森県

そういえばすかいらーくなんかも名前を変えてセルフサービスのファミレスをやっているみたいですね。とりあえず貧乏人には嬉しいことです。

◆雪なんか大嫌いだぁ！　河内　一真(18)広島県

2月12日，朝，外を見ると……東京は本当にすごかったです。

◆足にデキモノができたので医者にいくと液体チッソでジューッとやられた。あれは絶対拷問だ。痛かった。おまけにしばらくはまともに歩けない。編集部の方も一度経験してください。　仲村　正聡(23)大阪府

頼まれてもいやです。

◆HDドライブか北海道旅行か悩んだあげく，なかば強制的に旅行費用となった私の貯金。それというのも我が家におけるX68000の地位は非常に低く，たとえ私自身の貯金であっても貢ぐことはできないからだ。ごめんね，X68000Com

▲岡村　直也　兵庫県
姉弟4コマですが，今回は2本立。いかなる大団円を迎えるのでしょうか。突然現れたこの赤ちゃんはいったい……。右へ続く。

▲岡村　直也　兵庫県
知らないってことはとっても怖い。悪気がないだけに……。とにかく岡村さんご苦労さまでした。次のシリーズも期待してますね。

pactちゃん。　　　　　　　松尾　絢(17)長崎県

ここは家庭内でのX68000地位向上計画を立ててみてはいかがですか。でも実用ソフトは……。

◆熱くなりすぎちまったぜ！　すみません，2月号の86ページに「熱いメッセージを……」とあったのでつい。えっ，温度のことではないのですか？　どうも浪人で1年無駄にしてしまって，ものも無駄にしがちです（次は人生か？）。決戦の日は近い。！　　村松　充志(19)福岡県

ハガキが届いてびっくり。なんと右隅が一部燃えていました。コメントの通り熱くなりすぎてしまったようです。

◆大宮にはSEGAのゲームセンターがあり，店の前に電光掲示板を置いています。その掲示板に「サムライスピリッツ　X68にまけるな」というメッセージがありました。ということは次はサムライスピリッツがX68000に……。がんばれ魔法さん。　　　　　　　波田　雅之(20)埼玉県

残念ながら次回作は違うようですが，龍虎の拳が出れば可能性があるかもしれませんね。

◆2月号の津村君へ，君はアニ・ソンをばかにしてるね。それはよくないとお兄さんは思うな。コンパをしているならその場の流れに自分の身をまかせなければダメです（某大学アニメ研の部長さんがんばって)。ちなみに私の持ち歌は「男のポリシー」(「乙女のポリシー」の替え歌）ほかには北島三郎の「がまん坂」，アカペラでは「江戸の黒豹」なんかを歌っています。そうそう，皆さんもう飲みましたか？　ITOENの「ナタデココinおしるこ」けっこういけますよ（ただのおしること変わらんが……)。「ナタデココ」って昔「ココナッツゼリー」って名前で売られてましたよね。知ってます？

千装　茂夫(21)埼玉県

お酒の席では周りの人の迷惑にならないように気をつけてくださいね。どこにキレそうな人がいるかわかりませんから……。

◆2月号の椎名さんと竹原さんほか関心のある方々へ。さっそく1ケース買ってみたところ銀のエンゼルが2枚入っていました。上から見る方式では濃いのが1つ(はずれ)，薄いのが2つあって両方とも銀のエンゼルでした。金はなしです。確か手前のほうにあったと思います。味はピーナッツです。開いたチョコボールがあと18箱もある。こんなことでいいのか？　次はキャラメルだ！　　　　古橋　康宏(18)栃木県

あまったチョコボールは編集部に送ってくれれば食べてあげましょう。向かいのコンビニではチョコボールがいちばん上の棚に載っていたので上から見比べることができませんでした。残念。

◆とき卵を加えても許せるのは塩ラーメンだけではないかと，ふと思うのでした。

吉田　務(21)大阪府

おじやにもかかせないと思うのですが。

◆1月15，16日とセンター試験をしてきました。やってすぐの感触としてはまあまあかと思っていたのに，自己採点してみてショックを受けました。あのときもっとやっていればといつもながらの後悔をしてしまい，自分がいやになりました。思えばいままでの人生，後悔の山，ここぞというときにいつも失敗する中途半端な自分の生き方が無性に憎らしい。志望を下げたくないので90%浪人決定です（別にそれほど高い志望ではないですが……)。とりあえず10%を2次試験にかけてみます。　間渕　繁紀(18)静岡県

2次の結果もそろそろ出たころでしょうか。結果がどちらにしても，これからの人生が後悔をしない人生でありますように。

▲岩瀬　貴代美　福岡県
それにしてもどちらもとっても可愛いですね。実際のところ，どちらかのイラストに似てたりするんでしょうか。岡村さん教えてくださいね。

▲玉野　健一　奈良県
姉弟4コマの最終回を記念してというわけではないですが，常連さんより岡村さんのお姉さんの結婚式のお祝いが届いていますので紹介します。

◆あー，その辺に油田でもないかなあ。あったら受験なんて相手にしないのにな。

菊地　賢一(18)茨城県

油田が出たらすぐにでも連絡ください。

◆1月号の伊藤さんの話を読んで，さっそくCD音質良化をしようとした矢先，友人Aから「CD凍死」の話を聞きました。CDを冷やしたことにより音が割れてしまったそうです。私も実際に実験をしてみようと思いますが，なにせ相手がCDでは犠牲が大きいのでフロッピーで試してみようと思います。CRCエラーが直るかも。

杉田　瑞樹(18)新潟県

本当に試された方がいたんですね。ごめんなさい。責任持てないとはいったもののちょっと心苦しいです。

◆書店で目の前にあるのになかなか気配をさとらせないOh!Xは雑誌界の忍者といえよう。

松居　啓樹(18)富山県

そうか，本屋で最近見かけないという人も本当は気配を察知してくれていないだけかもしれない。

◆父親が業務用のビデオカメラを買おうと企んでいます。価格はフル装備のクラウンと同じくらいでしょうか。しかし俺は(ひょっとしたら)浪人するかもしれんというのになにを考えてんでしょう？　　　　　　金子　直史(17)新潟県

やっぱりなにか撮るんでしょうね。でもなにを……。

◆眠い，まるで受験生のような生活だ。だれかこの山積みの仕事をやってくれ。

内藤　陽一(27)東京都

ついでに私の仕事も……。

◆タクシーにぶつけられてしまった。幸いケガはなかったけどそれにしても本当に久しぶりに心の底から怒ってしまった。ドライバーの皆さんバックするときは後ろに注意してくださいね。

主藤　二裕(26)福岡県

バックするとき以外も気をつけてくださいね。

◆「ニケ」はアップルシード，「クロービス」はドミニオンという，どちらも士郎正宗という人が書いている漫画のキャラクタです（このてのハガキは63枚くらいきているはず)。

能登　康彰(22)北海道

どうもありがとうございます。ほかにも何枚かいただきましたが63枚ではありませんでした。念のため。

◆なんと会社をヤメて独立してしまいました。このまま食えなくなって死んだらどうしよう。オーエンお願いします。佐藤　伸一(29)岡山県

ファイト！

◆成人式の日にディスコに行った（私は24歳なのであまり意味のない日なのだが)。なかでは神主さんがお立ち台の上で客におはらいをしていた。2日後の深夜，その様子がテレビで放送されていた。なんだかディスコに行くたびに予想外のことが起きるので編集部の皆さんいっしょに行きませんか？　　中原　弘喜(24)神奈川県

某ディスコではお立ち台がなくなりましたが次はなにを考えているのでしょうか。

◆とりあえず会社は年を越すことができて，私も社内でのリストラには引っかからなかったが(同期もちらほら引っかかり……)，その代わり長期出向となってしまいました。ははは……はぁ。なんせ同じ埼玉でも電車で行くとドアTOドアで2時間30分強だぞ。いいかげんにせい！しかし次の仕事がないからなあ。う～ん考えるなあ……どうしよう？　　丹治　誠(25)埼玉県

そしてまた今日も会社へ……考えすぎて眠れなくならないようにしてくださいね。

◆近所の三毛猫は私が起きると庭にくる。「そんな目で俺を見るな。2浪の俺を責めるな」とたんに餌をねだりにきた猫に話しかける。友達の少ない私（涙)。　　　猪狩　友則(20)千葉県

猫は本当に餌をねだりにきているだけなのでしょうか……。

◆「猫はコタツでまるくなる」といいますが，ウチの猫は息苦しいのか顔と手をコタツから出して寝ています。　中島　民哉(23)埼玉県

その可愛い姿が目に浮かぶようです。

◆北海道の渡辺さん，レッズのサテライト（2軍のこと）は武南高校に負けたんですよ。いろいろ理由はあったみたいですが高校生にプロが負けるなんて……。今年は12チーム。目標は現状維持か（ため息）。　三浦　貴至(22)埼玉県

その件については，漫画のネタになっていました。そのオチはレッズがやるべきことを首脳陣に聞いたところ全力（1軍総動員）で武南高校に勝つ……。

◆なにげなくいつものようにX68000を使ってたら，いきなりドライブ0から異常音が聞こえた。何度やってもイジェクト用のモータからいやな音が聞こえる。こいつはついに壊れたかと作業をやめて1日ほっておいたらいきなり直ってた。いったいなんだったんだ？

伊与田　円良(21)東京都

働きすぎでちょっと休みたくなったのかもしれませんね。たまにはそんな日があってもいい……よくないって。

◆いつも発売日を忘れないように買っています。大学の生協ではすぐに売り切れるか，立ち読みでボロボロになってしまいます。

喜多　清高(24)兵庫県

立ち読みでボロボロになったOh!Xの運命はいかに。

◆筋肉質の人のためのCG作成支援ツール“マッチョエール”近日発売（ウソ）！

伊藤　直也(23)静岡県

たとえ発売されても個人的には使いたくないですね。サンプル画面はもちろん……。

◆冬道を運転するドライバーの皆さんは気をつけたほうがいいです。特に凍った路面では。僕の父親は「峠道でスピンしながら坂を下っていってガードレールでやっと止まった」といっていたので……。　大島　大介(17)北海道

お父さまはご無事だったのでしょうか？もう春ですが，皆さんも名残雪には気をつけてくださいね。

◆クリスマスイブの夜，歩いて友人の家まで行く途中に，道沿いに怪しいセンサーのようなものが5～6台立ててありました。マイクのような形をしたそれは多摩川上空の1点に向けられているようでした。そうか，彼はきっと多摩川沿いにやってくるんだな。その夜はずっとハンダづけで明けてしまいましたが，鈴の音を聞いたような気がします。　小林　宏昭(20)東京都

ロマンチックですね。でも我が家には来てくれなかったみたいです。子供がいないせいでしょうか……。

◆酔って帰ってきて，冷蔵庫にあった透明の液体を思いっきり飲みました。べにばなオイルでした。そのあとはもう……ウッ。もうペットボトルにはだまされないぞ。

佐川　正人(24)東京都

想像するだけで気持ちが悪くなりそうです。ゴマ油だったら許せたかも……。そんなことはないですね。酔ったときは気をつけます。

◆今年もらった年賀状は2枚。でもお年玉つき年賀ハガキで生まれて初めて当たりました（切手シートですが）。いきなり今年の運を使いはたしたような気がします。どうしましょう？

福田　浩人(23)新潟県

今年はとってもついているんじゃないですか？

◆X68000はともかく，PM4：00に寝ている彼女を起こす方法を教えて。　和田　智(18)岩手県

彼女と同じ生活スタイルにすれば問題ないのでは。でもPM4：00から朝まで起きないとしたら……。

◆MO買ったらもう大変（笑）。

志摩　憲(20)大阪府

座布団は取り上げてしまいましょう。

◆テレビでマヨネーズの一気飲みをやっていた。「げげっ，気持ちわる～」と思いながらも今年の忘年会で使えるかもと思ってしまう自分が悲しい。　藤原　常雅(23)神奈川県

すでに忘年会のことを考えているとは去年の忘年会に苦労したんですね。

◆うちの鶏は名前を呼ぶと寄ってきます。背中をなでても逃げません。卵も産むけど餌代のほうが高くついています。でも可愛いからいいや。

下田　達也(26)三重県

すっかり家族の一員ですね。でもその卵はやっぱり胃の中ですか。

◆私はマユ毛が薄くなって困っています。床屋に行って顔を剃ってもらっていると，マユ毛を剃らずに終わってしまう自分が悲しくてなりません。ヒゲは剃っていると濃くなるといいますが，マユ毛はどうなんでしょうか。剃ってみようかと思うのですが，そのまま毛が生えてこなかったらどうしようと思い実行できずにいます。皆さん，なにかいい知恵をください。

清野　一男(23)山形県

とりあえず剃ってみて生えてこなかったら化粧をしてごまかす……というわけにもいきませんね。

◆18歳になってからパチンコによくいくようになったが玉が出ない。財布から金が出ていくだけでOh!Xを買うお金もなくなってしまいそうです。やはり修行あるのみなのでしょうか。

渡辺　治男(18)東京都

あきらめが肝心ともいいます。

◆車が壊れて（14万円），スキー板（6万円）を買ってスキー（9万円）に行った。それくらいしかお金を使ったことが思い当たらないのに，もうボーナスがない。不思議だ。

横田　晶持(22)愛知県

友人とお酒を飲みにいったり，なんやかんやと知らないうちに使ってしまったのでは……そうでなければ不況でボーナスが少なかったのでしょうか。

◆「うる星やつら」がBSで再放送，嬉しい。これでLD-BOXを買わなくてもすみそう。さらに「あしたへフリーキック」もBSでやってくれる。年齢とともにオタク度が深まっていくような……。8mmVTRがもう1台必要になってしまった。雑誌が違うか。　沼　圭司(25)静岡県

好きなものはしようがないですよね。

◆いま，時代劇が面白い。といっても水戸黄門や遠山の金さんみたいな安易な勧善懲悪モノはだめだ。鬼平犯科帳や八丁堀捕物ばなしのような深い人間ドラマは下手な映画顔負けの内容である。最近始まった父子鷹もなかなかよさそうだ。ぜひ一度ごらんあれ。P.S.別にサムライスピリッツに感化されたわけじゃないぞ。もっと前から鬼平犯科帳見てたもん。

木村　奨(21)兵庫県

勧善懲悪モノもそれなりに味があると思うんですけど。

◆ある雑誌のバックナンバーを注文しようと本屋へ行った（Oh!Xは毎月買っているからその必要はない）。するとその本屋ではいっさい注文は受け付けていないそうなのだ。そんな本屋，あっていいのか？

赤松　宏章(22)大阪府

雑誌のバックナンバーだといろいろ難しいのかもしれませんね。書籍などだと大丈夫だと思うんですが……。

◆通信で他人のプロフィールを見るとたいがい

◀武田　正道　兵庫県
お金があればいいモノができるわけではありませんが，ないと少しつらいですよね。メモリは湯水のようだし，ほかにも……。

◀伊藤　浩克　香川県
気持ちよさそうに寝ていますが，このままだと風邪をひいてしまいそうですね。右上の小さな女の子は妖精でしょうか。よだれがちょっと心配。

パソコンの環境が書いてある。それを見ていると自分の環境が普通かそれ以下に思えてしょうがない。とことん拡張してやる！ MOもう１台とGバイトクラスのHDドライブ，FPU，イメージスキャナ，イメージユニットII……全財産投入して……その金で国民機とか買うつもりはまったくないです。　福永　浩司(22)大阪府

通信をしている人の普通の環境ってすごそうですね。

◆初めてスキーをしました。安いツアーで北志賀へ行ったのです。パラレルもどきができるようになったので次が楽しみです。ホテルの方にはチェックアウトしてから４時間近くロビーのソファに居座ってカップラーメンをすすりながらUNOをして，ご迷惑をおかけしました。私たち６名は世間の白い目にびくともせず，さぞいやな客だったでしょう。　服部　直幸(20)広島県

憎まれっ子世にはばかるといいますから。

◆某「○c fan」の創刊記念プレゼントで，「悪魔城ドラキュラ」と「出たな！ツインビー」が当たってしまいました。X68000ユーザーはこんな雑誌読まないだろうからと思って応募したのですが，本当に当たるとは。こういう「Windows系雑誌のX68000関連のプレゼント」は狙い目かも。　二村　直広(18)岐阜県

滅多にあることではないですが，一度に２本というのはすごい。

◆MOを買った。不思議とそれに見合うようにデータ量が増えている。　野崎　哲也(19)大阪府

あるものは使ってしまうということですね。

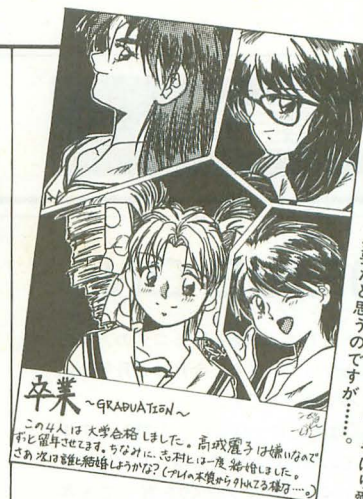
◆受験が終わって１年が過ぎ，大学生活をしておりますが工学部のレポートって数式が多くてワープロが使えません。皆さん手書きでがんばってはるんでしょうか。バックナンバーをひっくり返していたら，1991年８月号にTeXのことが載っているじゃあないですか！　これって通信やってないと入手できないんですか（しかし，理系大学生の友「TeX」とは……）。　古川　亨(19)兵庫県

弊社よりX68K Programing Seriesの３冊目として発売予定があります。よろしく。

◆スキーへ行ってきました。しかし，その日は猛吹雪で視界がきかないったらありゃしない。吹雪だからいつもより寒くて滑ってないと体が温まらない。それだけ多く滑れてよかったけど次の日は筋肉痛でした(笑)。今度は天気がいいときに行きたいです。　阿部　学(21)埼玉県

リフトが止まらない程度の吹雪でよかったですね。

◆家で職場でX68000とともにがんばっていますが，最近のMacintoshの大攻勢に多勢に無勢でどうしようもありません。職場ではMacintoshのネットワークが構築されます。X68000もOS-9を使えば相当なパワーを発揮するのではと思っています。そんなときのモニタ募集。使わせてくれ〜。　平山　謙司(43)福岡県

結果はどうであれ，がんばってください。

◀溝畑　知幸　兵庫県
卒業に出てくる女の子５人……あれ，高城がいないんですね。嫌いだということですが，やはり愛のある教育が必要だと思うのですが……。

◆とにかくつらい毎日が続いている。X68000 ACEがX68000PROが……懐かしの機種になりつつある今日この頃。SMC-777を触り始めた私はおじさんなのであろうか。LOGOはいい。とにかくプログラムを作り，脳の老化を防ぐ私は愚かでしょうか。妻もそろそろやめたら……とつぶやいています。　松本　祥司(32)東京都

楽しければいいんではないでしょうか？

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。
●紹介を希望されるサークルは必ず会誌の見本を送ってください。

売ります

★カラーイメージスキャナ「CZ-8NSI」を50,000円以上で売ります。箱，マニュアル，付属品などあります。往復ハガキに電話番号を明記のうえ連絡してください。〒331　埼玉県大宮市大成町2-319-2　野崎　徹(20)

★HAL研究所のファインスキャナ「HGS-68」を19,000円（送料別）で売ります。箱，説明書などあります。計測技研の数値演算プロセッサ「KGB-X68PRK-10」を27,000円（送料別）で売ります。箱，説明書，付属品あります。連絡は往復ハガキでお願いいたします。〒215　神奈川県川崎市麻生区岡上234-1 タカラハイツ201号　小林　徳光(24)

★システムサコムのSCSIボード「SX-68SC」を10,000円で売ります。ロジテックのSCSIハードディスクドライブ「LHD-FM200E」（200Ｍバイト）をケーブルつきで30,000円で売ります。どちらも箱なしですが説明書はあります。完動です。送料は別ですが，セットで購入される方は40,000円（送料込み）です。〒781-11　高知県土佐市宇佐町宇佐1945-4　坂本　知之(24)

★ローランドの音源モジュール「CM-32P」と「CM-32L」をセットで55,000円で売ります。ばらでも売りますが，セットの方優先です。値引き可。連絡は官製ハガキでお願いします。〒419-01　静岡県田方郡函南町上沢435-193　野畑　智也(17)

★エニックスのSCSIハードディスクドライブ「EFX-100B」（100Ｍバイト）を25,000円（送料込み）で売ります。箱，マニュアル，付属品などあります。１年ほどの使用で動作音がうるさいほかは完動品です。連絡は往復ハガキでお願いします。〒098-44　北海道天塩郡豊富町瑞穂南　佐藤　正年(35)

★アイテックのハードディスクドライブ「TX-80」（SASI，SCSIモード切り替え可能）を15,000円で売ります。箱，説明書，付属品あります。連絡は往復ハガキでお願いいたします。〒811-03　福岡県福岡市東区志賀島1276-21　高木　宣博(23)

★カラーイメージユニット「CZ-6VT1」を送料込みで25,000円で売ります。箱はありませんが説明書はあります。連絡は官製ハガキか往復ハガキでお願いします。〒214　神奈川県川崎市多摩区菅仙谷2-8-19 パーセク稲田堤201号　黒瀬　光男(25)

買います

★アイテックの「TX-80」（SASI対応，色は黒希望）を20,000円で買います。連絡は官製ハガキでお願いします。〒157　東京都世田谷区北烏山4-31-10　桜井　暢(39)

★X68000CompactXVI用２Ｍバイト増設RAMボード「CZ-6BE2D」と２Ｍバイト増設RAM「CZ-6BE2B」をセットで40,000円以下で買います。「CZ-6BE2B」が２枚のときは，60,000円以下で買います。完動品で付属品と説明書があれば箱はなくても構いません。送料込みでできるだけ安く売ってくださる方を優先します。連絡は官製ハガキでお願いします。〒934　富山県新湊市殿村133　戸倉　康博(21)

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々のご意見を紹介しています。今月は2月号の内容に関するレポートです。

●X-BASICは入門用ではない。入門から実用まですべての領域において素晴らしい完成度をもっていると思う。X68000を買って5年間，まともに使えるのはX-BASICだけだ。ただ，できることの幅は，ほかのパソコンのBASICとは比べものにならない。同人ゲームソフトのなかにもBASICをコンパイルしただけで十分に遊べるゲームができている。

そして，Cライクな使用方法やCG，音楽，計算，ファイル操作などが簡単な命令でできる手軽さが素晴らしい。

原田　謙(19)　X68000　PRO　石川県

●特集のタイトルを見て「重そうやな」と反射的に思ったのは私だけでしょうか？　ここまでX-BASICにこだわるのは「お手軽」というのがコンピュータに不可欠な要素なのでしょう。正直いって「お手軽」以外，長所のない言語だと私は思っています。

ただ「開発用言語」として「C言語としてコンパイルできる」というのは，ほかに例をみない利点だと思います。逆にコンパイルできるという利点の陰でインタプリタ本体がおろそかになっているような気はします。もう少し速くてもバチは当たらないと思うのですが。なんだかんだいってもBASICはインタプリタですからね。

中矢　史朗(23)　X68030, X68000 ACE-HD, PC-386P　愛媛県

●特集の「ショートプロのテクニックを盗め」では，BASICのプログラムをいつも以上に詳しく解説していて，わかりやすくてよかったです。連載でもあれくらいやってほしい。

八亀　桂一(19)　X68000 PRO　神奈川県

●「ストリートファイターⅡダッシュ」に「餓狼伝説2」。最近のX68000のソフトの出来のスゴさは本当に目をみはるものがある。個人的には「餓狼伝説2」に大賞をあげたいところだが，どうなるのか予想もつかない。

しかし，音楽部門は「悪魔城ドラキュラ」で決まりであろう。さすがコナミである。もし，「悪魔城ドラキュラ」の音楽のよさがわかっていない人がいたとしたら，よほどひねくれた人か，SC-55版を聴いたことがない人のどちらかだろう。あなたが後者だとしたら，悪いことはいわない，SC-55を買ってでも聴くのだ。そうしないと損をするぞ！

吉岡　洋明(20)　X68000 PRO，PC-8801 MA，FM-NEW7　埼玉県

●「ワンチップIC工作入門」を読んで，やはりパソコンを使ってロボットを操作，一度はやってみたいですね。これをキーボード操作ではなく無線操作にし，あらかじめ行動パターンをプログラムにしておく。実際に「ロボットコンストラクションR.C.」のようなことができるようになるかもしれませんね。

森崎　剛(21)　X68000 XVI，PC-9801 RX21　広島県

●最近ポリゴンが気になり「ハードコア3Dエクスタシー」に目を通しています。

さて，少林寺拳法初段，現正道会館練習生の私にいわせていただくならば,「バーチャファイター」のアクションパターンはよくできています。重心，体重移動，腰のひねり……よく研究していると思います。

そこで「SLASH」というシステムが公開されているのは非常に嬉しいことです。いまの自分の実力ではなにもできないかもしれませんが，「SLASH」のおかげで見えてきた可能性にドキドキしています。丹さん，横内さんがんばってください。

橋本　和典(27)　X68000　XVI, LC520　東京都

●このところ「DōGA CGアニメーション講座ver.2.50」が元気ですね。EPA 2の話が始まってから，おもしろいと思って読んでいます。今回いちばんよかったのは,「森山さんのお勧めアニメ作品」です。いやぁ片寄っています。『迷宮物語』の『走る男』のスピード感は，確かに素晴らしいものがありました。でも，私は『工事中止命令』のほうが好きです。『パトレイバー』の劇場版にも同じ血が流れていますね。あ，アニメの話になってしまいました。

野原　賢次(32)　X68000 ACE-HD，X1 turbo model30　埼玉県

ごめんなさいのコーナー

2月号　ワンチップIC工作入門

P.97　2段目にある「ほとんど無条件に，75××とか76××という……」とあるのは「78××とか79××」の間違いでした。

P.99　図8左側のジョイスティックポートの番号は上から順に5，1，2，3，4，9が正しいものです。また，240Ωと300Ωの抵抗の位置が入れ替わっています。

P.100　図9に270Ωは，正しくは240Ωです。また，⑥とあるのは⑨の間違いです。ご迷惑をおかけしたことをお詫びいたします。

3月号　Oh!Xreader'sぎゃらりぃ

P.17　印刷工程でのミスにより，2枚ほど名前が抜けていました。右上が藤沢篤(奈良県)さんで，その左斜め下が加藤隆（佐賀県）さんの作品です。申し訳ありませんでした。

3月号　マッドストーカーX68

P.31　欄外の工画堂スタジオの名前が間違っていました。関係者の方々に重ね重ねご迷惑をおかけしたことをお詫びいたします。

3月号　ひなまつりPRO-68K

・Z's-EX&MATIER-EX

付録ディスクに収録されていたzs_ex.sysを以下のように修正してください。

1　先頭の行を512(改行)にしてください。
2　バージョン　→　：バージョン
3　zseyes.x　→　exeyes.x
4　lipe　→　lupe

・AMI SYSTEM

サンプルとして収録されたBOXinBOXがコンパイルオプションの関係で実行できない機種があります。Cコンパイラをお持ちの方は各自でソースをコンパイルしてくださるようお願いします。また，差分については来月号の付録ディスクで収録する予定ですが，容量の関係で割愛させていただく場合もあります。不手際がありましたことをお詫びします。

3月号　(で)のショートプロぱーてぃ

P.70　ZTEMPO.Sの画面が正しく表示されません。203，205行を以下のように変更してください。

```
.dc.w '現在値          ださい
↑ (＋増)'
.dc.w '設定範囲 ーで～択して ださい
← (ー減)'
```

どうも申し訳ありませんでした。

バグに関するお問い合わせは
☎03(5642)8182(直通)
月～金曜日16：00～18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。

また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

笑ってプログラミング！SX-WINDOW

▶今回の特集は予告から一転のSX-WINDOWとなりました。周辺機器の特集を期待していた方はごめんなさい。それにしても，今月号が出る頃には「EGWord SX-68K」やSX-WINDOW開発キット「Workroom SX-68K」，SX-WINDOW開発キット用サポートツール集が発売されていることでしょう。今月紹介できなかったものは追ってレポートします。

なにやら急にSX-WINDOWの周囲が騒がしくなってきました。先月号の付録に収録された「SX-BASIC」「MTEXT」なども使ってみましたでしょうか。「SX-BASIC」についてはまだまだデバッグの必要があります。しかし，「Workroom SX-68K」や「SX-BASIC」の登場でSX-WINDOW上のプログラミングの可能性が広がったことも事実です。SX-WINDOWでこんなことやあんなこと，やってみたいと思うことがあったらバグを恐れずにとりあえず動かしてみましょう。

これを機会に，あなたもメモリをいっぱい積んでプラットフォームにSX-WINDOWを使ってみませんか？

▶1993年GAME OF THE YEARの発表が行われました。皆さんの予想はどうだったでしょうか。当たりましたか。結果は……でした(本文を見てね)。そして，今回の発表の時点ですでに1994年GAME OF THE YEARはスタートしています。ゲーム好きの方は，来年の応募に備えてしっかり遊んでくださいね。

▶来月号は毎年恒例の「言わせてくれなくちゃだワ」です。すでにアンケートのほうも続々と寄せられています。今年もどんな意見が出てくるのか，いまから楽しみです。そして，どうもお待たせしました。3月号に続いての付録ディスクには，今度こそ「SLASHver.2.0」が収録される予定です。ほかにもゲームやツール，各種データなど内容は盛りだくさんになりそうです。

▶「X68000マシン語プログラミング入門」「ファイル共有の実験と実践」は著者多忙のため，今月はお休みとさせてもらいました。

投稿応募要領

- 原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。
- プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたテープ（ディスケット）を添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。
- ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。
- 投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒103 東京都中央区日本橋浜町3-42-3
ソフトバンク出版部
Oh!X「㊁㊀㊇㊈」係

SHIFT・BREAK

▶私の住んでいるアパートは窓もドアも隙間だらけで冬場は冷蔵庫いらずだが，寒くて布団から一歩も出られないので，5,000円で安売りしていたセラミックファンヒーターを買ってみた。翌月，電気料金がもう1台セラミックファンヒーターを買えるくらい跳ね上がっていて，いまでは怖くてスイッチが入れられず歯をカチカチ鳴らしている始末。(H)

▶うちの「てくまく」がブートしなくなってしまった。こないだは，ほかの機種のハードディスクも壊れたし，某ノートパソコンのドライブも死んでいる。そういえば，最近，どうも帯電体質になっているらしくて，指先にバシバシ静電気を感じる。もしかして原因はこれか？ 次は編集室のマシンに危害を及ぼさないか，ちょっと心配……。(で)

▶原稿を書いてる最中にカゼを引いてしまった。たいてい寝冷えが引き金になるのだが，今回は違う。うどん食ってて七味にせき込んだのが原因らしい(笑うな！)。その日，鼻の奥が痛かったのが一晩明けたら腫れていたというわけ。中耳炎に点耳薬をし，点鼻薬とイソジンで消毒し，疲れ目に目薬をして原稿を書く。ああ外用薬ジャンキー生活。(E.K.)

▶今月はX68000よりもPC-9821を触っている時間のほうが多かった。PC-9821ではFM音源6声，PSG3声，リズム音源6声，PCM音源(44.1kHz)が2声と豪華絢爛な音源が標準搭載されている。しかし，これを制御できるソフトウェアがろくにないというのが悲しい。というわけでPC-9821用のZ-MUSICを誰か作らない？ いや，ほんと，マジで。(善)

▶突然普通免許を取ろうと思い立ったら，5月に法改正で教習所は駆け込み組で超満員。1カ月近く経つのにまだ運転できない始末。果たして法改正までに間に合うのだろうか？ 社会人になってから取るのは金額的，体力的にきつい。学生のうちに合宿で取っとけばよかった。しかしどうやったら交通関係の法律があそこまで複雑怪奇になるんだ？(A.T.)

▶マックワールドエキスポ/東京へ行った。朝日新聞に取り上げられたせいもあってか（アダルトソフトコーナーを隔離して，入口に警備員を立たせたって記事だ)，最終日の混雑ぶりはお祭り騒ぎのデパート状態。メディアとしてのパソコンがいよいよ認知されたわけだ。その影で，テクポリとポプコムが休刊したそうな。ああ。時代ってややこしい。(K)

▶MRIの結果も異常なし。なぜかその直後，風邪で寝込んでしまった。もう一度別の内科にかかる。2週間後，もう風邪は治りましたよといわれたが体温が下がらない。普通の生活に戻れば下がりますよといわれて2週間。いまだに体温は37度を越えている。体調も思わしくないし，僕の体はいったいどうなったのか。少し心配だ。(以下次号のKO)

▶冬季オリンピックが終わった。今回もいろいろあったが，ジャンプの起源は処刑の一種であると何かで読んだ。スポーツとして認められれば，日に当たった場所だが，それ以前の各スポーツの起源には結構暗い過去があるのかもしれない。そういえば，ジャンプで処刑された人が，万が一に無事に飛ぶことができたら罪を許してもらえたのだろうか？(高)

▶ハードディスクをフォーマットした。システムをバージョンアップした。ついでに整理整頓が行われる(筈だった)。ほったらかして遊んでるうちに，仕事が忙しくなった。結局，机のまわりや部屋の中とおんなじで，ちらかったまま使っている。あたしにはWHEREコマンドという強い味方があるもんね。実生活でもこんなものないかなあ……。(ふ)

▶最近お気に入りの商品名。それは「人造バラン」だ。この「人造バラン」というのは，よく弁当に入っている緑色のギザギザしたやつ。ゴミ同然の存在なのに結構な名前をもっているじゃん，しかもなんとなくショッカーのできそこないみたいでかわいい，というのがお気に入りの理由。さて，現在新モデラを坪井氏が制作中。来月号に間に合うか？(J)

▶「冬季びっくり人間大会」と馬鹿にしてても目にするとなぜか観入ってしまう(フィギュアだけだが)。動機や背景はともかく一所懸命やってる姿は見てて気持ちいい。それはともかく噂の新ヤマトが現れた。完結編で「ヤマトの眠りが永遠であることを願う」というナレーションに，しみじみと「願うのぅ」と呟いた，とある先輩の姿が目に浮かぶ。(U)

▶ハーディングはぜんぜん仕上がってなかったので問題外だったが，一方のケリガンのあまりにもリスクを感じさせない演技を見ていたら，彼女が妬まれるのもむりはないと思ってしまった。並みの精神力じゃないよ。被害にあったのは気の毒だけど，悲劇のヒロインなんていったら，ほかの選手たちがかわいそう。私はバイウルが勝ってほっとした。(T)

microOdyssey

最近，周囲で結婚をする友人が増えている。男性の場合は特に問題ないのだが，女性の場合，結婚後に話題にのぼると，つい旧姓で呼んでしまう。そういう経験はないだろうか？　きっと現在の姓よりも旧姓のほうを強く記憶しているからだろう。

では，記憶とはどういったものなのだろう。辞書で調べてみると「過去に経験したことや一度覚えたことを，時間がたったあとまでも大体その通りに思い出せること」とある。昔，記憶というものは，ろう板に鋭いものを押し当てたときに跡ができるようなものだといわれていた。そしてろう板は時がたつにつれて摩滅していき滑らかになる。その滑らかな状態が忘却である。確か，プラトンの説だったと思う。もし，その通りならば，私のろう板に跡をつける道具はずいぶん種類があるようだ。特に勉強に関する記憶はずいぶんなまくらなもので跡をつけたらしい。また，かのデカルトは著書「思索私記」のなかで，自分の記憶力の弱さを嘆いている。そこで彼は，記憶に頼らない疑いようのない前提をおいて，そこから論理的に演繹をするというやり方を確立した。同じ記憶力のなさを嘆くにしてもずいぶん違うものだ。

話がそれたが，冒頭で書いたような現象を，現代ではプライミング(priming)と呼ぶそうだ。これは，以前にある経験をした場合，再び同じか似たような状況にでくわすと，その経験か別の類似の経験が出やすいことをさす。ただその状況にでくわしたときにとる行動は意識的ではないのだ。ということは「覚えのない記憶」といったところだろうか。

もしかしたら，国会議員の方々やどこかの会社のお偉いさんが「身に覚えがないことです」というのも，案外本当なのかもしれない。つまり，あれだけお金のやりとりをしていれば「覚えのない記憶」にもとづいて無意識にやっているだけということだ。

話は変わって，少し古い本で岡嶋二人の「クラインの壺」(新潮社刊)がある。テーマにバーチャルリアリティ(以下VR)が取り上げられている。内容については本書を読んでもらいたいが，VRを体験できる機械を使っているうちに，自分が機械の中にいるのか外にいるのかわからなくなってしまう話である。

現実と混同してしまうほどのVRを体験できるものは現在まだないと思うが，できたとしたら，人はその体験で得た記憶を現実での体験とどう区別していくのだろう。VRを体験するときが意識のあるときなら問題はないが，無意識のときだったら現実の記憶と同じになってしまうのではないか。しかし，現実と区別する必要は実はないのかもしれない。そのとき自分が感じたことに間違いはないのだから，その記憶も嘘ではないはずだ。でも，VRで体験したことが「覚えのない記憶」として吸収され，無意識に現実の行動に影響を与えていたら結構楽しい(怖い？)かもしれない。

そういえば，編集長がＪリーグのチケットがずいぶん取れたと喜んでいた。これがしばらく続くとそのうち「チケット取るなんて簡単だよ」なんていいだすかもしれない。そういい始めたら，私のチケットもお願いしておこう。でも，いつのことやら。　(高)

1994年5月号4月18日(月)発売

こいのぼりPRO-68K

・SLASH ver.2.0
・SX-BASIC(バグ修正版)
・パズルゲームPUSH BON！
・その他，各種プログラム，データなど

新製品紹介
Workroom SX-68K/開発キットサポートツール集

特別付録　5"2HDディスク　　予価800円

バックナンバー常備店

東京	神保町	三省堂神田本店5F 03(3233)3312
	〃	書泉ブックマートBI 03(3294)0011
	〃	書泉グランデ5F 03(3295)0011
	秋葉原	T-ZONE 7Fブックゾーン 03(3257)2660
	八重洲	八重洲ブックセンター3F 03(3281)1811
	新宿	紀伊国屋書店本店 03(3354)0131
	高田馬場	未来堂書店 03(3209)0656
	渋谷	大盛堂書店 03(3463)0511
	池袋	旭屋書店池袋店 03(3986)0311
	八王子	くまざわ書店八王子本店 0426(25)1201
神奈川	厚木	有隣堂厚木店 0462(23)4111
	平塚	文教堂四の宮店 0463(54)2880
千葉	柏	新星堂カルチェ5 0471(64)8551
	船橋	リブロ船橋店 0474(25)0111
	〃	芳林堂書店津田沼店 0474(78)3737
	千葉	多田屋千葉セントラルプラザ店 0472(24)1333
埼玉	川越	黒田書店 0492(25)3138
	川口	岩渕書店 0482(52)2190
茨城	水戸	川又書店駅前店 0292(31)0102
大阪	北区	旭屋書店本店 06(313)1191
	都島区	駸々堂京橋店 06(353)2413
京都	中京区	オーム社書店 075(221)0280
愛知	名古屋	三省堂名古屋店 052(562)0077
	〃	パソコンΣ上前津店 052(251)8334
	刈谷	三洋堂書店刈谷店 0566(24)1134
長野	飯田	平安堂飯田店 0265(24)4545
北海道	室蘭	室蘭工業大学生協 0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある『新規』『継続』のいずれかに○をつけ，必要事項を明記のうえ，郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので，大切に保管してください。なお，すでに定期購読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は，上記と同じ要領でお申し込みください。

基本的に，定期購読に関することは販売局で一括して行っています。住所変更など問題が生じた場合は，Oh!X編集部ではなくソフトバンク販売局へお問い合わせください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS(株)にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(3238)0700

 4月号
■1994年4月1日発行　定価600円(本体583円)
■発行人　橋本五郎
■編集人　稲葉俊夫
■発売元　ソフトバンク株式会社
■出版事業部　〒103 東京都中央区日本橋浜町3-42-3
Oh!X編集部　☎03(5642)8122
販売局　☎03(5642)8100 FAX 03(5641)3424
広告局　☎03(5642)8111
■印　刷　凸版印刷株式会社

バックナンバー案内

BACK ISSUES

ここには1993年4月号から1994年3月号までをご紹介しました。現在1992年6，7，12，1993年6〜12，1994年1〜3月号の在庫がございます。バックナンバーはお近くの書店にご注文ください。定期購読の申し込み方法は148ページを参照してください。

1993

4月号（品切れ）
特集　X68第7世代へ

連載　DōGA CGアニメーション講座/マシン語プログラミング
響子 in CGわ〜るど/ショートプロ/よいこのSX-WINDOW
ハード工作/吾輩はX68000である/Computer Music入門

● 決定！ 1992年GAME OF THE YEAR
● 名作ゲーム再遊記

LIVE in '93　FIGHTMAN/ミンキーモモより 愛しのマーシカ
THE SOFTOUCH　スターフォース/元朝秘史　他
全機種共通システム　シューティングゲームコアシステム作成法(2)

5月号（品切れ）
特集　襲撃！ SX-WINDOW
第8回　言わせてくれなくちゃだワ

連載　DōGA CGアニメーション講座/ANOTHER CG WORLD
響子 in CGわ〜るど/ショートプロ/大人のためのX68000
ハード工作/吾輩はX68000である/Computer Music入門

● X68030へのソフトウェア対応について

LIVE in '93　MAGICAL SOUND SHOWER/もう笑うしかない　他
THE SOFTOUCH　エトワールプリンセス/メガロマニア　他
全機種共通システム　シューティングゲームコアシステム作成法(3)

6月号
創刊11周年特別企画　確率遊技シミュレーション

連載　DōGA CGアニメーション講座/こちらシステムX探偵事務所
響子 in CGわ〜るど/ショートプロ/大人のためのX68000
ハード工作/吾輩はX68000である/Computer Music入門

● 新製品紹介　SC-55mkII

LIVE in '93　ストリートファイターIIより 春麗のテーマ/BAY YARD/LOVE&CHAIN
THE SOFTOUCH　餓狼伝説/信長の野望・覇王伝　他
全機種共通システム　REVERSI

7月号
特集　席巻するローテク文明

連載　DōGA CGアニメーション講座/こちらシステムX探偵事務所
響子 in CGわ〜るど/ショートプロ/マシン語プログラミング
ハード工作/吾輩はX68000である/Computer Music入門

● 新製品紹介　ドローイングパット33070&MATIER

LIVE in '93　Midnight Circle/今日の日はさようなら/赤い靴
THE SOFTOUCH　悪魔城ドラキュラ/リブルラブル/大航海時代II/銀河英雄伝説III/幻影都市/ヴェルスナーグ戦乱
全機種共通システム　MSX用S-OS "SWORD"

8月号
特集　C言語実践的入門

連載　DōGA CGアニメーション講座/こちらシステムX探偵事務所
響子 in CGわ〜るど/Computer Music入門/大人のためのX68000
吾輩はX68000である/ショートプロ/ANOTHER CG WORLD

● 特別企画　夏真っ盛り，アマチュアリズムのX68000

LIVE in '93　SPLASH WAVE
THE SOFTOUCH　悪魔城ドラキュラ/リブルラブル/餓狼伝説/ロボットコンストラクションR.C./Winning Post
全機種共通システム　MACINTO-C再掲載

9月号
特集　光学式磁気円盤MO

連載　DōGA CGアニメーション講座/こちらシステムX探偵事務所
響子 in CGわ〜るど/ショートプロ/大人のためのX68000
ハード工作/Computer Music入門/ANOTHER CG WORLD

● 新製品紹介　OS-9/X68030

LIVE in '93　ファイナルファンタジーVのテーマ/銀河鉄道999/アルスラーン戦記IIより　汗血公路/ちょうちょ
THE SOFTOUCH　悪魔城ドラキュラ/コットン/ダーク・オデッセイ 他
全機種共通システム　7並べ/SLANG再々掲載

10月号
特別企画　秋祭りPRO-68K

連載　ハードコア3D/Computer Music入門/マシン語プログラミング
DōGA CGアニメーション講座/こちらシステムX探偵事務所
響子 in CGわ〜るど/ショートプロ/吾輩はX68000である

● 特別付録　秋祭りPRO-68K（5"2HD）
● SCSIパックンTOWER JACK

LIVE in '93　未来予想図II/OutRunより　PASSING BREEZE
THE SOFTOUCH　コットン/The World of X68000/あにまーじゃんV3
全機種共通システム　シューティングゲームコアシステム作成法(4)

11月号
特集　ポリゴナイザSLASHの活用

連載　ハードコア3D/Computer Music入門/ファイル共有の実験と実践
こちらシステムX探偵事務所/目指せジョイスティックの星
響子 in CGわ〜るど/ショートプロ/大人のためのX68000

● 新製品紹介　Easydraw SX-68K
OS-9 Ultra C/Technical Tool Kit

LIVE in '93　渚のアデリーヌ/エロティカ・セブン
THE SOFTOUCH　ぶたさん/ダイアット・ヴァークス
全機種共通システム　S-OSで学ぶZ80マシン語講座(1)

12月号
特集　古今東西ゲーム議論

連載　ハードコア3D/マシン語プログラミング/響子 in CGわ〜るど
DōGA CGアニメーション講座/こちらシステムX探偵事務所
ショートプロ/Computer Music入門/ファイル共有の実験と実践

● 新製品紹介　MATIER ver.2.0
C Compiler PRO-68K ver.2.1 NEW KIT

LIVE in '93　クリスマス・イブ/星に願いを
THE SOFTOUCH　ネメシス'90改/項劉記/スーパーリアル麻雀PII&PIII
全機種共通システム　エディタアセンブラREDA再掲載

1994

1月号
特集　Z-MUSICシステムver.2.0

連載　ハードコア3D/ゲーム作りのKNOW HOW/響子 in CGわ〜るど
DōGA CGアニメーション講座/こちらシステムX探偵事務所
ショートプロ/Computer Music入門/ファイル共有の実験と実践

● 特別企画　ANOTHER CG WORLD in Hong Kong

LIVE in '94　LAST WAVE/スターウォーズ/明日への扉/夢路より　他
THE SOFTOUCH　ストリートファイターIIダッシュ/餓狼伝説2/ドラゴンバスター/X68000傑作ゲーム選
全機種共通システム　S-OSで学ぶZ80マシン語講座(2)

2月号
特集　X-BASICとグラフィック

連載　ハードコア3D/ワンチップIC/響子 in CGわ〜るど
DōGA CGアニメーション講座/こちらシステムX探偵事務所
ショートプロ/Computer Music入門/ANOTHER CG WORLD

● 新製品紹介　ハイパーピクセルワークス

LIVE in '94　ランス3/新宿駅，巣鴨駅の発車メロディ/ピコー・ソング
THE SOFTOUCH　キーパー/マッドストーカーX68/餓狼伝説2　他
全機種共通システム　S-OSで学ぶZ80マシン語講座(3)
YGCSver.0.20リファレンスマニュアル

3月号
特別企画　ひなまつりPRO-68K

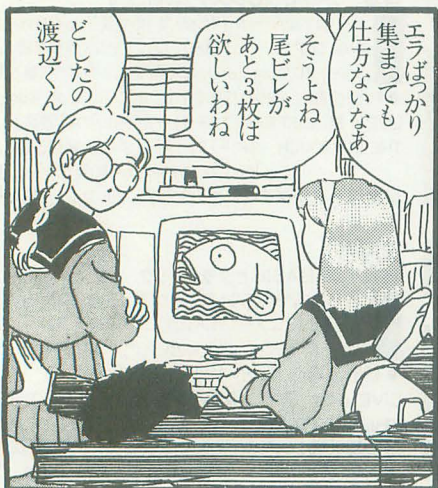
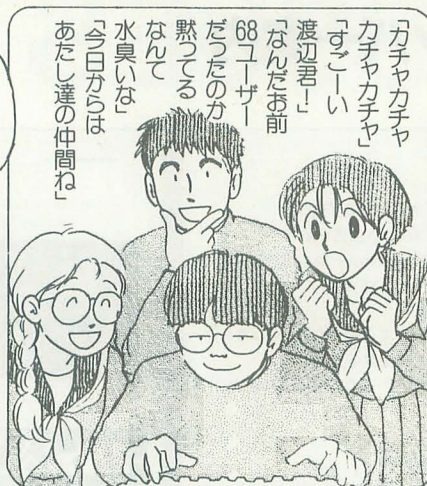
連載　ハードコア3D/マシン語プログラミング/ゲーム作りのKNOW HOW
DōGA CGアニメーション講座/こちらシステムX探偵事務所
ショートプロ/響子 in CGわ〜るど/ファイル共有の実験と実践

● 特別付録 ひなまつりPRO-68K（5"2HD）
● 新製品紹介 ビデオPC for X680x0

LIVE in '94　THEME FROM WINNING RUN/スターフォースアレンジ版
THE SOFTOUCH　卒業/マッドストーカーX68/B-Field! 他
全機種共通システム　S-OSで学ぶZ80マシン語講座(4)

満開の電子ちゃん

作・え 岡村祭

別冊四号は、只今発売中です。なお別冊五号は、5月発売予定です。

講読方法：定期購読もしくはソフトベンダーTAKERUでお買い求めいただけます。
★定期購読の場合＝第71号(94年4月号)より6ヶ月分の場合、6,500円(送料サービス、消費税込)を、現金書留または郵便振替で下記の宛先へお送り下さい。
　現金書留の場合：〒171 東京都豊島区長崎1-28-23 Muse西池袋2F ㈱満開製作所
　郵便振替の場合：東京 5－362847 ㈱満開製作所
●ご注文の際は、郵便番号・住所・氏名・電話番号を忘れずに記入して下さい。
●3.5インチディスク版をご希望の方は、「3.5インチ版」とご指定下さい。
●新規購読の方は「新規」と明記して下さい。なお、特に購読開始号のご指定がない場合は既刊の最新号からお送りいたします。
●製品の性格上返品には応じられませんが、お申し出があれば定期購読を解約し残金をお返しします。
★TAKERUでお求めの場合＝1部につき1,200円(消費税込)です。
●定期購読版と内容が一部異なる場合があります。御了承下さい。
●お問い合わせ先 TEL(03)3554－9282 (月～金 午前11時～午後6時)
(なお、定期購読版のバックナンバーについては定期購読の方のみご注文を承ります)

Vol.76(94年9月号)より、毎月2枚組 1,500円(本体1,456円)に価格改定されます。今後とも、電脳倶楽部を宜しくお願いいたします。
また、別冊(季刊)の定期購読も開始いたしました。「別冊○号より」(既刊は4号です)と明記の上、年間購読料8,000円(本体7,767円)をご送金ください。

株式会社 BLUE SKYは、X68000/30でラップトップパソコンライクなレジューム機能を使うことが出来る様になる“レジューム・マスター”を5月下旬に発売する予定である。

X68000/30のメイン・メモリー、グラフィック・メモリー、PCG、テキスト・メモリーなどをHDDにデータとして記録し、任意の時にそれを展開して電源スイッチを切った時の状態を、再現する事が出来る。

コンピュータの電源スイッチを切るとそれまでのデータは失われる、今までは電源スイッチを切る前にデータをセーブするなどの、煩雑な手順を踏む必要があった。
“レジューム・マスター”を使うと、それらの作業なしで電源スイッチを切っても、それまでのデータが保存される。
地震などの様に停電が予想され、緊急にデータの待避をしたい等の時には、1～2分でデータのバックアップが出来て非常に便利である。

しかし、下記の場合は対応していない。
FM音源を使用するソフトは、FM音源の各レジスタをデータ展開後に再設定する必要がある。
SX-WINDOWなどの様に自前で電源OFFルーチンを持っているソフトではレジューム機能が働かない。
標準仕様ではない外部増設機器はサポートしていない。
S-RAMを使用するソフトとは共存出来ない。
なお、HDD内に使用するコンピュータのメイン・メモリーの容量＋約1.1メガバイトの空き容量が必要である。

★発売記念として5月18日迄に直接当社にご予約の方に限り、標準価格1万8千円のところ1万5千円にさせて頂きます。
使用するコンピュータのメイン・メモリーの容量により、1～12メガバイト用の12種類有ります。
フロッピーのサイズも忘れずにお書き添え下さい。

レジューム・マスター　1～12メガバイト用
5″・3.5″2HD　各**18,000**円（消費税別）

■商品名・機種名・メディア名・住所・氏名・電話番号を明記の上、現金書留または郵便為替にてお申し込み下さい。（送料無料）

BLUE SKY Co.
株式会社 BLUE SKY
〒411 静岡県三島市加茂16-4 ☎0559-72-6710

ズバリご奉仕

P&Aならではの 新品パソコン5年保証

《業界№1の"P&Aメンテナンスサポート"》

最高の保証システム

① 業界最長の新品パソコン5年保証
（※モニター・プリンター3年間保証!! ※一部商品は除きます。）
② 中古パソコンの1年間保証（※モニター・プリンター6ヶ月間保証!!）
③ 初期不良交換期間3ヶ月（※新品商品に限らせていただきます。）
④ 永久買取保証
⑤ 配達日の指定OK!!（土曜・日曜・祭日もOK!!）
⑥ 夜間配達もOK!!（※PM6:00〜PM8:00の間 ※一部地域は除きます。）

便利でお得な支払いシステム

① 翌月一括払い手数料無料（ご利用下さい。）
② 業界№1の低金利!!
③ 月々の支払いは￥1,000より
④ 9ヶ月先からのスキップ払いOK!!
⑤ 84回までの分割、ボーナス併用OK!!
⑥ カレッジクレジット
⑦ ステップアップクレジット
⑧ ボーナスだけで10回払いOK!!
⑨ 現金一括支払いOK!!
⑩ 商品到着払いOK!!（代引き手数料が必要になります。10万円まで900円）
（※商品・金額ご確認の上、銀行振込・現金書留にてご入金下さい。）

●法人向け リースシステム
業務に最適なシステムを構築します。
損金処理が可能なリース契約をどうぞ。

周辺機器コーナー （送料￥1,000・消費税別）

カラーイメージスキャナ
■JX-325X
定価￥190,000
特価**￥143,000**

カラーイメージジェット
■IO-735X-B
定価￥248,000
特価**￥128,000**

ビデオスキャナー
■CZ-6VS1
定価￥178,000
特価**￥135,000**

FDD（5インチ×2基）
■CZ-6FD5
定価￥99,800
P&A超特価
￥49,800

プリンター（ケーブル用紙付）
- MJ-500V2（エプソン）………特価￥51,300
- MJ-1000V2（〃）………特価￥71,300
- VP-1047PC（〃）………特価￥49,000
- BJ-220JC（キヤノン）………特価￥61,300
- BJ-10V Lite（〃）………特価￥36,000
- BJ-15V PRO（〃）………特価￥54,000
- LBP-A404GII（〃）………特価￥99,500
- BJC-820J（〃）………特価￥154,300
- JET505J PLUS（YHP）………特価￥53,300

光磁気ディスク（X68000用）
■CS-M120（コパル）
●ケーブル、ターミネータ付￥178,000
特価**￥96,500**
■LMO-FMX330
●ケーブル、ターミネータ付￥178,000
特価**￥135,000**

- CZ-6TU……定価￥33,100▶特価￥23,900
- CZ-8NM3……定価￥9,800▶特価￥7,200
- CZ-8NT1……定価￥13,800▶特価￥10,000
- CZ-6BE2A……定価￥59,800▶特価￥42,800
- CZ-6BE2B……定価￥54,800▶特価￥39,300
- CZ-6BE2D……定価￥54,800▶特価￥39,300
- SH-6BF1……定価￥49,800▶特価￥36,500
- CZ-6BP1……定価￥79,800▶特価￥57,000
- CZ-6BM1……定価￥26,800▶特価￥19,300
- CZ-6SD1……定価￥44,800▶特価￥32,500
- SH-6BN1……定価￥29,800▶特価￥21,800
- CZ-6BV1……定価￥21,000▶特価￥15,200
- CZ-6BC1……定価￥79,800▶特価￥57,000
- SH-6BG1……定価￥59,800▶特価￥43,800
- CZ-6PV1……定価￥198,000▶特価￥142,000
- CZ-6BS1……定価￥29,800▶特価￥21,500
- CZ-8NJ2……定価￥23,800▶特価￥17,500
- CZ-6BL2……定価￥298,000▶特価￥214,000
- CZ-6CS1（674C用）定価￥12,000▶特価￥8,900
- CZ-68HA……▶特価￥91,000
- CZ-6CR1（RGBケーブル）・定価￥4,500▶特価￥3,600
- CZ6CT1（テレビコントロール）・定価￥5,500▶特価￥4,400
- CZ-6BP2……定価￥45,800▶特価￥33,300
- CZ-5MP1（X68030用）・定価￥54,800▶特価￥42,000

■システムサコムボード
- SX-68MII（MIDI）
定価￥19,800▶特価￥13,500
- SX-68SC（SCSI）
定価￥26,800▶特価￥17,500

（X68030用）
- CZ-5BE4
定価￥54,800 ▶￥42,000
- CZ-5ME4
定価￥49,800 ▶￥38,000

X68000用ソフトコーナー （送料￥700・消費税別）

- Z's STAFF PRO68K Ver.3.0（ツァイト）
……定価￥58,000▶特価￥37,500
- Z's TRIPHONYデジタルクラフト（ツァイト）
……定価￥39,800▶特価￥27,000
- マジックパレット（ミュージカルプラン）
……定価￥19,800▶特価￥14,200
- たーみのる2（SPS）
……定価￥17,800▶特価￥13,000
- Mu-1 Super（サンワード）
……定価￥39,800▶特価￥28,500
- サイクロンEXPRESS α68
……定価￥98,000▶特価￥69,000
- Video PC for X680X0（マイクロウェアシステムズ）
……定価￥58,000▶特価￥46,400
- X WINDOWS V.11.5（マイクロウェアシステムズ）
……定価￥30,000▶特価￥25,500
- Double Book IN（計測技研）
……定価￥12,800▶特価￥9,600
- OS-9/X68030 V.2.4.5（マイクロウェアシステムズ）
……定価￥25,000▶特価￥19,900
- C&Professional Pack V.3.2（マイクロウェアジャパン）
……定価￥80,000▶特価￥57,800
- マチエール Ver.2.0
……定価￥39,800▶特価￥28,800
- CZ-213MSD MUSIC PRO68K
……定価￥18,800▶特価￥13,200
- CZ-214MSD SOUND PRO68K
……定価￥15,800▶特価￥11,300
- CZ-215MSD Sampling PRO68K
……定価￥17,800▶特価￥12,500
- CZ-220BSD DATA PRO68K
……定価￥58,000▶特価￥40,000
- CZ-225BSV Multiword Ver.2.0
……定価￥32,000▶特価￥23,000
- CZ-243BSD CYBERNOTE PRO68K
……定価￥19,800▶特価￥15,000
- CZ-247MSD MUSIC PRO68K（MIDI）
……定価￥28,800▶特価￥20,500
- CZ-249GSD CANVAS PRO68K
……定価￥29,800▶特価￥22,000
- CZ-251BSD Hyperword
……定価￥39,800▶特価￥29,400
- CZ-253BSD CARD PRO68K Ver.2.0
……定価￥29,800▶特価￥22,700
- CZ-257CSD Communication PRO68K Ver.2.0
……定価￥19,800▶特価￥15,300
- CZ-258BSD Teleportion PRO68K
……定価￥22,800▶特価￥16,900
- CZ-261MSD MUSICstudio PRO68K Ver.2.0
……定価￥28,800▶特価￥21,200
- CZ-263GWD Easypaint SX-68K
……定価￥12,800▶特価￥9,800
- CZ-264GWD Easydraw SX-68K
……定価￥19,800▶特価￥15,300
- CZ-265HSD NewPrintShop Ver.2.0
……定価￥20,000▶特価￥15,400
- CZ-266BSD PressConductor PRO68K
……定価￥28,800▶特価￥22,000
- CZ-267BSD CHART PRO68K
……定価￥38,000▶特価￥29,800
- CZ-271BWD EG-Word
……定価￥59,800▶特価￥44,900
- CZ-272CWD Communication SX68K
……定価￥19,800▶特価￥14,500
- CZ-275MWD SOUND SX68K
……定価￥15,800▶特価￥11,500
- CZ-284SSD OS-9/X68000 Ver.2.4
……定価￥35,800▶特価￥25,600
- CZ-286BSD BUSINESS PRO68K
……定価￥28,000▶特価￥20,500
- CZ-288LWD開発キット（workroom）
……定価￥39,800▶特価￥29,700
- CZ-289TWD 開発キット用ツール集
……定価￥12,800▶特価￥9,600
- CZ-290TWD SX-WINDOW ディスクアクセサリー集
……定価￥14,800▶特価￥11,500
- CZ-294SS（5"）/SSC（3.5"）SX-WINDOW Ver.3.0
……定価￥19,800▶特価￥15,200
- CZ-295LSD C-Compiler PRO68K Ver.2.1 NEW KIT
……定価￥44,800▶特価￥32,500

☆ゲームソフト25%OFF OK!!（一部ソフト除く）

全国通販

★頭金なし！
★即日発送

●お近くの方はお立寄り下さい。専門係員が説明いたします。
●本体単品で特価で受付します。詳しくは電話にてお問合せ下さい。
●ビジネスソフト定価の20%引きOK！TEL下さい。

※お支払いは、便利な商品到着払い〈手数料（10万円まで900円）要〉をご利用下さい。

P&A特選 今月の中古特選品

- CZ-600C…￥55,000
- CZ-601C…￥65,000
- CZ-611C…￥70,000
- CZ-652C…￥75,000
- CZ-612C…￥95,000
- CZ-603C…￥85,000
- CZ-653C…￥78,000
- CZ-612C…￥90,000
- CZ-623C…￥110,000
- CZ-674C…￥108,000
- CZ-634C…￥130,000
- CZ-644C…￥178,000

※上記は単品価格、モニター別売。

新古品 限定	新古品 限定	新古品 限定
●CZ-674CH ●CZ-608DH **￥138,000**	●CZ-634CTN（チタン）（中古） ●CZ-613D（グレー）（新品） **￥190,000** （モニターをCZ-614TN（チタン）に変更の場合￥20,000加算）	●CZ-644CTN ●CZ-604DB **￥228,000**
中古品 ●CZ-674CH ●68000専用モニター付 **￥128,000**	中古品 ●CZ-634CTN ●68000専用モニター付 **￥158,000**	中古品 ●CZ-644CTN ●68000専用モニター付 **￥198,000**

中古・高価現金買取り／下取りOK!!

■まずはお電話下さい。
下取り専用
買取り電話 ▶ ☎03-3651-1884 FAX.03-3651-0141
■下取り・買取りで、お急ぎの方は、直接当社に来店、または宅急便にてお送りください。

買取り価格…完動品・箱／マニュアル／付属品の価格です。

●下取りの場合…価格は常に変動していますので査定額を電話で確認してください。（差額は、P&A超低金利クレジットをご利用ください。）
●買取りの場合…現品が着き次第、2日以内に高価買取金額を連絡し、振込み、又は書留でお送り致します。
●近郊の方はP&A本店に直接お持ちください。即金にて￥1,000,000までお支払い致します。

●最新の在庫情報・価格はお電話にてお問い合せください。
●買い取りのみ、または、中古品どうしの交換も致します。詳しくは電話にて、お問い合せください。
●価格は変動する場合もございますので、ご注文の際には必ず在庫をご確認ください。
●本商品の掲載の商品の価格については、消費税は、含まれておりません。
●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せください。

P&A特選パソコンラック&OAチェアー （消費税込み）（送料無料、離島を除く）

※全機種→キャスター付
※上から2番目棚板移動可能（4/5段）
※フレーム色：4段→黒、3/5段→ホワイト

通信販売お申し込みのご案内

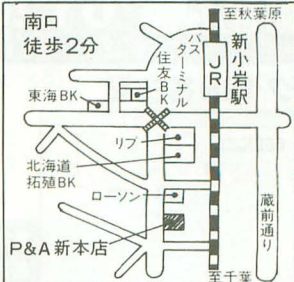
〔現金一括でお申し込みの方〕
●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで現金書留でお送りください。（プリンター・フロッピーの場合、本体使用機種名を明記のこと）

〔クレジットでお申し込みの方〕
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。●1回〜84回払いまで出来ます。但し、1回のお支払い額は￥1,000円以上。

〔銀行振込でお申し込みの方〕
●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。（電信扱いでお振込み下さい。）

〔振込先〕 さくら銀行 新小岩支店
当座預金 2408626 （株）ピー・アンド・エー

超低金利クレジット率

回数	3	6	10	12	15	24	36	48	60	72
手数料	2.9	3.9	4.9	5.1	8.1	10.9	15.2	20.0	25.8	33.6

P&A
株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目2番地20号
●営業時間：AM10：00〜PM7：00 日・祭：AM10：00〜PM6：00
☎03-3651-0148（代）
FAX.03-3651-0141
●定休日／毎週水曜日

●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上お申し込み下さい。詳しくは、お電話でお問い合せ下さい。

安いのに親切 **TSUKUMO**

※4/18からはゴールデン・ウィーク・セールが始まります!!

フレッシュ・スタート・セール開催中!! 4/17まで

ツクモと一緒に新しいスタートを切りましょう!!

CF ON AIR中!!
「安い春見つけた!」

ツクモがあなたのフレッシュライフ(新生活)を応援します。
〜イメージガール越智静香さんも元気に春を彩ってます!!〜入学祝はパソコンがいいね!

★価格はどれも、ど〜んと安いツクモ特価だから、欲しいものがたくさん買えるネ!!★

X680x0シリーズ本体

大好評に付き、特別セール延長!!

なんと 67%OFFです。

CZ-674C-H(X68000 CompactXVI)・・・**超特価¥98,000**

TS-XFDCAを使えば、5インチモデルX68000シリーズを外部ドライブとして使用可能!

是非、2台目のマシンとしてどうぞ!

おすすめの組み合わせ!!

CZ-500C-B........定価¥398,000
240MBハードディスク.....サービス
ツクモ特価¥318,000

CZ-300C-B.........定価¥388,000
TS-XFDCA.........定価¥ 9,800
ツクモ特価¥295,000

満開製作所の商品も取扱中!

X68000 CompactXVI 24MHz改

RED ZONE....................ツクモ特価¥160,000
RED ZONE + MK-FD1...ツクモ特価¥180,000

満開製外付け5インチFDD

MK-FD1..........................ツクモ特価¥39,800
MK-FD1(カラーリングモデル)......ツクモ特価¥44,800

ディスプレイも特別価格にて提供中!

CZ-607D(14型カラーディスプレイテレビ)..........ツクモ特価¥ 60,000
CZ-608D(14型カラーディスプレイ)..........ツクモ特価¥ 69,000
CZ-615D(15型カラーディスプレイテレビ)..........ツクモ特価¥132,000
CZ-621D(21型カラーディスプレイ)..........ツクモ特価¥125,000

大容量記憶装置

SCSIボードが必要な場合にはセット価格に¥22,000加算となります。

ハードディスク

品名	価格
120MBハードディスク	ツクモ特価¥39,800〜
200MBハードディスク	ツクモ特価¥42,800〜
240MBハードディスク	ツクモ特価¥49,800〜
340MBハードディスク	ツクモ特価¥69,800〜
540MBハードディスク	ツクモ特価¥99,800〜

X68000/030シリーズ用RAMボード

品名	ツクモ特価
SH-6BE1-1ME(CZ-600C専用)	¥10,800
PIO-6BE1-AE(ACE/PRO/PRO2シリーズ用)	¥10,800
PIO-6BE2-2ME(拡張スロット用)	¥22,800
PIO-6BE4-4ME(拡張スロット用)	¥38,800
SH-5BE4-8M(X68030シリーズ用)	¥44,800
CZ-6BE2A(XVI専用)	¥42,500
CZ-6BE2D(CompactXVI専用)	¥29,800
TS-6BE2B(CZ-6BE2A/D用拡張RAM)	¥29,800
XSIMM10(8MB搭載仕様,拡張スロット用)	¥53,800

MO特選セット

Panasonic
LF-3100B 定価¥178,000
MOメディア 付属
SCSIケーブル サービス
ツクモ特価¥99,800

コパル
CS-M120PX(ブラック) 定価¥178,000
SCSIケーブル サービス
MOメディア サービス
ターミネータ サービス
ツクモ特価 ¥118,000

SONY
RMO-S360 定価¥169,000
MOメディア 付属
SCSIケーブル サービス
ツクモ特価 ¥128,000

「コレが欲しい!」とお決まりになったら、今すぐ!!
お電話一本!お気軽にどうぞ
受注専用フリーダイヤル **0120-377-999**
通販センター・・・03-3251-9911 商品についてのお問い合わせは各店または通販へ。

クレジット払い
月々¥3,000以上の均等払いも頭金なし。夏・冬ボーナス2回払いも受付中!

各種リース払い
くわしくは各店にお問い合わせ下さい。ケースに合わせてご相談承ります。

現金書留払い
〒101-91 東京都千代田区神田郵便局私書箱135号 ツクモ通販センター Oh!X係

カード払い(¥5,000以上)
通信販売でのご利用カード、ツクモグローバルカード、セントラル、ジャックス※ご本人様より電話で通販部へお申し込み下さい。

全国代金引換え配達
お申し込みはTEL03-3251-9911へお電話1本!配達日の指定もできます。

銀行振込払い
事前にTELでお届け先をご連絡下さい。三和銀行 秋葉原支店(普)1009939 ツクモデンキ

ツクモIN名古屋 (1号店 第一アメ横ビル内 / 2号店 第二アメ横ビル内)

1号店 担当 横山

2号店 担当 松原

名古屋1号店 TEL052(263)1655 休 毎週火曜日
名古屋2号店 TEL052(251)3399 休 毎週水曜日

ツクモIN札幌 (ツクモ札幌店 / DEPOツクモ2番街店)

札幌店 担当 田口

DEPO店 担当 鈴木

札幌店 TEL011(241)2299 休 毎週木曜日
DEPO店 TEL011(242)3199 休 毎週木曜日

業界No.1!!

12回払い、7.5%がナント6%に!

クレジット金利がこんなにお安くなりました!!~これで欲しかったパソコンが購入出来ます!!~

支払回数(回)	1	3	6	10	12	15	18	20	24	30	36	42	48	54	60
分割払い手数料率(%)	2.5	3.5	4.5	5.5	6	9	11.0	12	12.5	16.5	17.5	22	23	28.5	29.5

ツクモグローバルJCBカード登場!!

好評 入会者受付中!

18才以上なら 学生でもOK!

JCBならではの国内、海外サービスにツクモオリジナルの特典をプラス。お支払いはプランに合わせて、1回・2回・ボーナス一括・リボルビング払いから選べるのでとても便利!!ツクモ各店備え付けの入会申込書にてお申し込み下さい。詳しくはグローバル事務局03-3251-9898または各店へ。

★ジャックス・VISA・セントラル・マスターも取り扱っております。

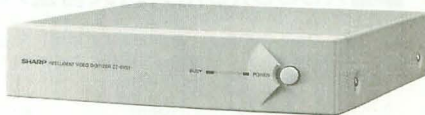
超低金利!夏・冬ボーナス二回払い受付中!詳しくは各店までお問い合わせ下さい。

動画を始めてみませんか?

ツクモ特価販売中!!

ビデオ入力ユニット CZ-6VS1

定価¥178,000

MC68020(25MHz)の32BitMPUを搭載し、SCSIインターフェイスを介してパソコンへデータを転送。動画・静止画を簡単に保存出来るアプリケーションソフト「ライブスキャン」を標準装備。1,677万色まで対応し、最大640×480ドットの高解像度で、高速取り込が可能です。

コンピュータアート

スーパーグラフィックツール NEW セット

その1.慣れてしまうとマウスがいらない

DrawingPad‥‥定価¥76,500
Matier Ver2.0‥‥定価¥39,800

合計定価¥116,300 **ツクモ特価¥88,000**

その2.ハイオリティなのにこんなに安い

BJC-600J‥‥定価¥120,000
プリンターケーブル‥‥定価¥4,800
Matier Ver2.0‥‥‥‥定価¥39,800

合計定価¥164,600 **ツクモ特価¥128,000**

プリンター

48ドットカラー熱転写プリンター
CZ-8PC5-BK **ツクモ特価¥39,800**

バブルジェットプリンター
BJ-10VLite(ケーブルセット) **ツクモ特価¥38,500**

NEW
カラーバブルジェットプリンター
BJC-600J
(ケーブルセット)
定価¥120,000
ツクモ特価¥99,800

カラーイメージスキャナー

CZ-8NS1 **ツクモ特価¥79,800**
JX-325X **ツクモ特価¥135,000**

MIDIインターフェース

CZ-6BM1A **ツクモ特価¥19,000**

ツクモオリジナルシリーズ

TS-3XRシリーズ X680x0用外付けドライブ

- 2DD/2HD/2HC/1.44MBフォーマット対応
- Human68K Ver.3.0以上が必要
- CompactXVI/68030用ケーブル付

TS-3XR1B 1ドライブ 定価¥33,800
ツクモ特価¥26,800

TS-3XR2B 2ドライブ 定価¥46,800
ツクモ特価¥36,800

限定販売中 (好評に付き、増産致しました。)

カラーイメージユニット接続ボックス
TS-VTBOX 定価¥19,800 **ツクモ特価¥17,800**

・CompactXVI/X68030シリーズにカラーイメージユニットを接続する為のアダプターです。

3月リリース予定!

Music Card for X680x0 (TS-6GM1) **発売記念特価¥39,800**

・MIDIボードにGM規格の音源を搭載しております。これ1枚で手軽にMIDIコンピュータミュージックが楽しめます。

MIDIコンピュータミュージック特選セット

Rolandセット

SC-55mkII 定価¥69,000
SX-68MII 定価¥19,800
Mu-1GS 定価¥28,000

合計定価¥116,800

→ **ツクモ特価 ¥92,000**

パソコン通信

モデム

AIWA	PV-AF144V5	ツクモ特価¥37,000
OMRON	MD144XT10V	ツクモ特価¥37,800
MicroCORE	MC1440FX	ツクモ特価¥34,800
Panasonic	TO-703B	ツクモ特価¥36,800

通信ソフト

たーみのる2 ……ツクモ特価¥13,000
Communication SX-68K ……ツクモ特価¥16,800

CD-ROMドライブ(2倍速)

ELECOM	ECD-250(TOSHIBAドライブ)	ツクモ特価¥47,800
東芝	XM-4100A(TOSHIBAドライブ)	ツクモ特価¥47,800
Logitec	LCD-500(SONYドライブ)	ツクモ特価¥56,800
SONY	CDU-7811(SONYドライブ)	ツクモ特価¥45,800
緑電子	CXA-301(NECドライブ)	ツクモ特価¥36,800

6連装CD-ROMドライブ

PIONEER	DRM-602X(2倍速)	ツクモ特価¥77,000
	DRM-604X(4倍速)	ツクモ特価¥178,000

CD-ROMドライバーソフト+SCSIケーブル ツクモ特価¥9,200

ソフトウェア

	ツクモ特価
OS-9/X68030 V2.4.5	¥20,000
Technical Tool Kit V.2.4.5	¥16,000
UltraC&Professional Pack V1.1	¥36,000
X Windows V11.5	¥24,000
SX-WINDOW Ver3.0システムキット	¥15,800
SX-WINDOWデスクアクセサリ集	¥11,800
C COMPILER PRO-68K Ver2.1NEWKIT	¥35,800
Easydraw SX-68K	¥15,800
Easypaint SX-68K	¥10,200
SOUND SX-68K	¥12,600
Communication SX-68K	¥16,800
Matier Ver2.0	¥29,800
CD-ROM Driver	¥4,800
SX-PhotoGallery	¥15,800
DoubleBookin'	¥12,800
EG Word SX-68K	¥47,800
Workroom SX-68K(SX-WINDOW開発キット)	¥31,800
開発キット用ツール集	¥10,200
倉庫番リベンジSX-68K	¥5,440

ツクモIN東京

営 平日 AM10:45~PM7:30 日・祝 AM10:15~PM7:00~ツクモ全店、3月末日まで無休営業となります~

ツクモパソコン本店II 4F

担当 荒井
TEL03 (3253) 1899 (直通)
ツクモパソコン本店II代表
TEL03 (3253) 4199 休毎週木曜日

ツクモニューセンター店

担当 沢栄
TEL03 (3251) 0987
休毎週木曜日
※下取り交換、中古販売も行っております。

各店、定休日が祝日と重なる場合は営業致します。

POLYPHON

X68000 サブMPUボード ～ポリフォン～

■POLYPHONはアクセラレータではありません！

POLYPHONはサブMPUボードです。アクセラレータと異なりメイン(本体)のMPUには干渉されません。従って、メインとは別のタスクとして処理できます。ですからPOLYPHON用のアプリケーション実行させながら、別のプログラムをX68000本体で実行するといったことも可能となります。ポリフォンシステムとの組み合わせにより、DoGA(REND.X)やGCC・HAS・HLKなどの実行ファイルもX68000本体と同時に別タスクとして動作可能。POLYPHON-24使用時にはパフォーマンスが約2.0～約2.4倍に向上します。

■POLYPHONはメモリボードにもなります

POLYPHON上にはサブMPUが使用する2MBの他にX68000本体用のメモリを最大8MB搭載できます(0MB/8MBモデルとして販売)。本体用メモリ部分は純正メモリボード同様に使用できます(サブ用メモリはどちらのモデルも2MBですが、こちらは増設できません)。

■POLYPHONはコプロボードにもなります

POLYPHONはコプロを装着することが出来ます(コプロ付モデルは装着済)。コプロ部分は純正互換ですので、FLOAT3などで簡単に利用することが出来ます(コプロ機能は本体用として機能します)。コプロ無モデル購入の方は、差額にてコプロのみの販売もします。

■POLYPHONはMIDIボードにもなります

POLYPHON上にはMIDIコネクタを装備(1IN/2OUT)しています。残念ながらこちらは純正非互換ですが、Z-MUSIC,MLD,RCシステムをはじめとする各種ミュージックドライバーもPOLYPHONのMIDI OUTをサポートしているので安心です。また、市販ソフトに関してはPOLYPHON-MIDI対応パッチを用意していますので、こちらを利用すれば問題なく利用できます。(パッチはPOLYPHONシステムディスクに付属)(市販ソフトでもZ-MUSIC対応ならば、Z-MUSICの差替えのみで動作します)

■本体にない付加機能も提供します

POLYPHONには本体にない機能としてステレオPCM機能を提供しています。POLYPHON上にステレオ出力端子を備え、高品質にPCMを再生します。

POLYPHON標準価格

POLYPHON	メインメモリ8MBモデル	¥85,000-(税別)
POLYPHON	メインメモリ8MBモデル(68881付)	¥95,000-(税別)
POLYPHON	メインメモリ0MBモデル	¥62,000-(税別)
POLYPHON	メインメモリ0MBモデル(68881付)	¥72,000-(税別)

遂にアップグレード開始

大変おまたせ致しました。POLYPHON-16からPOLYPHON-24へのアップグレードが始ります。詳しいアップグレード方法はDMにてご案内致しますので、そちらをご参照下さい。

POLYPHONシステムディスクのバージョンアップを受け付けています。随時最新の内容でお届けします。ご希望のユーザーは80円切手6枚を希望メディア(3.5"または5")を明記した上で、弊社まで送ってください。(ブランクディスク3枚と返送用切手でも可)(最新版は平成6年3月10日版)

■X680x0用外付大容量ハードディスク■

プログラム・音楽データ・画像データ…とハードディスクの足りない方にオススメ。フォーマット済のため、接続後にすぐ使用できます(パーティション分割する場合は、一旦領域解放し、再度領域を確保してください)。

1.3GB (D)	平均アクセスタイム10ms	¥153,000-	(**オススメ**)
1.8GB (Q)	平均アクセスタイム10ms	¥178,000-	
2.4GB (S)	平均アクセスタイム8ms	¥238,000-	
2.4GB (F)	平均アクセスタイム11.5ms	¥223,000-	
3.4GB (S)	平均アクセスタイム11ms	¥288,000-	
270MB (Q)	平均アクセスタイム10ms	¥49,800-	(**NEW！**)
Syquest 3.5" 270MBリムーバブル		お問い合せ下さい-	(**NEW！**)

QはQuantumドライブ使用　SはSeagateドライブ使用　FはFusitsuドライブ使用　DはDECドライブ使用
(容量はすべてアンフォーマット状態ですのでフォーマット後の容量は多少変わりますのでご了承ください)
すべてケーブル付。
その他の容量も取り扱っていますので、お問い合わせください。

お買求め・お問い合せは…

弊社製品は直販のみの販売でSHOPではお求めになれません。詳しい購入方法や細かい仕様などの資料を用意しておりますので、郵便番号・住所(都道府県からお願いします)・氏名を明記の上、ハガキにてご請求ください(代金を直接送らないで下さい)。

毎日沢山の資料請求のハガキが届いておりますが、配達先不明で返送されてくるものがあります。難しい文字には読み仮名を付けて下さい。

お願い…

最近電話番号間違いの苦情が多くなっております。弊社電話番号、弊社サポートネット電話番号前後へ電話する方がいらっしゃるようですが、該当番号者は大変迷惑しているとの苦情が入りました。他番号への電話はご遠慮下さる様お願い致します。

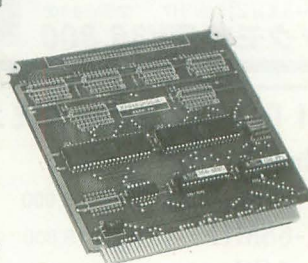
NEO COMPUTER SYSTEMS

株式会社ネオコンピュータシステム
120 東京都足立区綾瀬1-33-7-103
TEL 03-5680-7531(土・日・祝を除くAM10:00-PM4:00)
FAX 03-5680-7539(昨年よりFAX番号が変更になっています)
NET 03-5680-7533(サポート専用ネット)

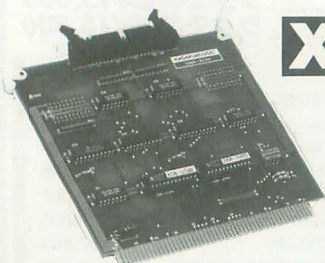
X68K-PPI 自作派御用達 8255コンパチボード

当社は博物館や科学館等の展示物(ハード・ソフト)を制作しています。この技術と経験からX68シリーズ用I/Fボード「X68K-PPI」を制作しました。グラフィックや音楽と同期してソレノイドやモーターを動かすのに必要なインターフェースボードとして作られたのが「X68K-PPI」です。

●48ビットI/Oボード。セミキット。●μPD71055(8255コンパチ)2個搭載。●入出力用バッファICを搭載できるエリアを用意。(8ビット×6個分)●X68030対応。●全回路図公開。使用しているGALの論理も公開。●定価22,000円(送料・税込み)

注意：本製品はセミキットです。入力出コネクターやバッファIC、プルアップ抵抗等は添付しておりません。ユーザーにて御用意お願いします。(山-FAP-60-07.02B等。)半田付け作業が必要です。

X68K-SCAN 電脳絵師に贈る スキャナボード

エプソンGTシリーズスキャナで高速入力を行うためのボードです。X680x0の優れたグラフィックエディター「マチエール」「Z's STAFF PRO-68K Ver, 3.0」で使えます。(添付ソフト使用時。)

●エプソンGTシリーズスキャナ用パラレルボード。●接続ケーブル付き完成品。●「マチエール」「Z's STAFF PRO-68K Ver, 3.0」でパラレル入力ができるようにするソフト添付。(5/3.5インチ同梱)●X68030対応●「マチエール」で512×512ドット6万5千色を1分強で入力。(X68030使用時。ちなみにRS-232C 19200bpsで7分17秒。当社測定)●対応スキャナ：エプソンGT-1000/4000/6000/6500/8000(GT-6500にはエプソンのシリアル・パラレルボードGT65RSPRBが必要です。)●全回路図公開。ソフトはソースも添付。コピーフリー。●増設プリンターポート/汎用パラレル入出力ポートとしてもお使い頂けます。●定価29,000円(送料・税込み)

注意：シャープ製パラレルボードCZ-6BN1との互換性はありません。「マチエール」は㈱サンワードの製品です。「Z's STAFF PRO-68K」は㈱ツァイトの製品です。

＝通信販売の方法＝

ご注文は、住所・氏名(会社名)・TEL・品名・個数を明記の上、郵便振替か現金書留にてお願い致します。入金確認後発送いたします。現金書留の場合はおつりのないようにお願いします。振替手数料・書留送料につきましてはお客様負担となります。(送料・消費税は代金に含む)その他技術的なご質問等FAX・郵便にて受付けております。

郵便振替：東京0-665905

株式会社 科学工芸研究所
〒164 東京都中野区本町5丁目14番23号
TEL.03(5385)4651　FAX.03(5385)4650

RPG幻想事典 戦士たちの時代

司史生·坂東いるか 共著　定価1,800円

ファンタジーRPGの背景である中世という時代の解説書。RPGの代表的キャラクターでもある戦士(騎士·兵士·武士)の活躍を中心にわかりやすくまとめました。平易な文体と緻密な史実の考証により、RPGファンの少年·少女から歴史読物ファンの大人まで、幅広く読める内容です。

ソフトバンク　スーパーガイドシリーズ

ファイヤーエムブレムスーパーガイド

Theスーパーファミコン編集部 編　定価780円

任天堂から発売されたスーパーファミコン用ソフト「ファイヤーエムブレム」の徹底ガイドブック。全44マップの解説および戦略に必要なアイテム·ユニットデータを完全にフォローしています。類書にはない登場キャラクター事典が付いています。

NetWare386/J リファレンス·マニュアル

池田安広 著　定価2,700円

本書は、NetWareユーザでMS-DOSコマンドの知識を踏まえていることを前提に、MS-DOSコマンドとの比較、過去の経験談を入れながらNetWareを解説しています。本書の構成は、1)MS-DOSとNetWareの環境の違い　2)リファレンスマニュアル　3)ログインスクリプト　4)APPEDIX(トピックス·ワークシート·Q&A)となっています。

すぐに使えるNetWare Lite/J

山口至 著　定価2,900円　3.5"2HDディスク付

小規模LANの代名詞とも言われる「NetWare Lite」をいかに設定したら、簡単かつ失敗しないかをわかりやすく、丁寧に解説しました。実際に運用していく際に便利なユーティリティ·ツールを付属の3.5"2HDディスクに収録しています。

一太郎 Ver.5 for Windows SUPER BOOK エントリーユーザ編

エクスメディア 著　定価2,000円

Windows上で一太郎をどのように動かせば良いのかを見開き単位でわかりやすく解説。本書はWindowsの基本操作に始まり、一太郎の特長を活かせるような使用方法を丁寧に具体例を挙げ示しています。

一太郎 Ver.5 for Windows SUPER BOOK パワーユーザ編

エクスメディア 著　定価2,000円

本書はWindowsと一太郎の両方をパワフルに楽々と使いこなしたいと希望するユーザ向けの一冊。精細な記述性と便覧性を両立させるため、見開き形式にまとめました。

一太郎 Ver.5 for Windows First Book

山口学 著　定価2,500円　3.5"2HDディスク1枚付

「一太郎はまったく初めて」という方でらくらく使いこなせるようになる格好の入門書。挨拶文から見積書、請求書まで、豊富なサンプルを収録したディスク付なので、ビジネスでも、ご家庭でもすぐに役立つ一冊です。

情報の共有化をめざす!

パソコンLANドキュメントシステム導入ガイド

小寺次夫 著　定価2,400円

パソコンLANで運用する文書データベースの導入によって実現する部門·部署レベルの情報共有システムの構築法を、身近で具体的な事例をもとに解説した、LAN導入担当者·管理者のための実践ガイドです。オフィス情報の統合化を目的としたLAN導入にまつわる様々な失敗例や落とし穴も示しながら、効率的で無理なく運用できる情報共有システム構築の考え方とノウハウをわかりやすく説明しています。

定価はすべて税込です。お近くの書店でお求めください。

好評発売中！

スケジュール管理ソフト

DoubleBookin'

標準価格 **¥12,800**

SX-WINDOW だから、いっしょに暮らせる

フリーソフトウェアセレクションVol.2
収録ソフト/データ募集中!!

〆切は4月30日！！

みなさんがつちかってきたX680x0文化が、CD-ROMに凝集します。
詳しい募集要綱、および応募フォーマットは、主要パソコン通信ネットワークの掲示版等でご覧になれます。ネットワーカー以外の方は、当社フリーソフトウェアセレクション担当まで電話やFAXにてご請求ください。

TEL(0286)38-0301　FAX(0286)38-0305

SX-WINDOW。そのマルチタスク環境では、ユーザーの作業環境は一変します。より自由に、感性のおもむくままに。SX-WINDOWとDoubleBookin'の組み合わせで、あなたの暮しが、仕事が、変わります。
DoubleBookin'は、あたらしい考え方にもとづいたスケジュール管理ソフトです。

●多角的な予定設定/表示

予定はカレンダー、デイリー、グラフの3つのウィンドウで確認することができ、多角的に検討することができます。
何日もにまたがる長期の予定にも対応。短期の予定とあわせて、日常生活に即したスケジュール設定が可能です。

●マルチタスクをいかした豊富なイベント

予定を設定した時刻をメッセージやアラームで知らせるのはあたりまえ。DoubleBookin'は、SX-WINDOWのマルチタスク環境をいかして、様々なイベントを起こすことができます。音楽を演奏したり、テレビ画面に切り替えたり、シャーペンやEasy drawを起動したり…など、思うままのライフスタイル設計を可能にします。

●モジュール追加で成長し続けます

予定に設定したイベントの種類は、外部モジュールを追加することによってさらに増やすことができます。
今後新しいアプリケーションや周辺機器が登場した場合でも、DoubleBookin'はそれらを取り込んで成長し続けます。

●電子手帳やシステム手帳とリンク

シャープ製電子手帳(DB-Z以上)とのスケジュールのやりとりを可能にしたほか、カレンダーやスケジュールのリフィル印刷もサポート。自宅やオフィスに縛られることなく、幅広いフィールドで活用できます。

●アンフィニーシステム社製MIC-68Kにも正式対応

スケジュールに応じて、赤外線で家電製品をコントロールできます。

ユーザウィンドウ
ディスク容量が許す限り、何人のスケジュールでも登録可能。

カレンダーウィンドウ
スケジュールの設定状況が、ひと目で確認できるカレンダー表示。ここが作業の中心です。

デイリーウィンドウ
カレンダー上の日付をダブルクリックすると、一日のスケジュールを詳しく表示します。

編集ウィンドウ
スケジュールの入力はとてもかんたん。
イベントアイコンを選んで予定の内容を記入するだけ。

スクープ！

X68030用　MC68040搭載アクセラレータ

040turbo

4月発売予定・価格未定

パソコン通信で話題沸騰のX68030用アクセラレータ「*040turbo*」を、当社から販売することが決定!
*040turbo*は、X68030(CZ-500C/510C)のMPUソケットに差す、68040搭載のドータボード形式のアクセラレータです。040のクロックは25MHz(無改造のX68030の場合)。強力なキャッシュの効果でパフォーマンスは劇的に向上します(下表参照)。
詳細は次号以降で!

プログラム	68030(キャッシュオン)	68030(キャッシュオフ)	68040(コピーバック)	68040(ライトスルー)	68040(キャッシュオフ)
dhry.x(ドライストン)	6002.4	4230.1	22624.4	14285.7	3232.1
キャンバス.X("草原.JPG"使用)	約17秒	約22秒	約6秒	約8秒	約23秒

(当社調べ)

発売中
CD-ROM Driver Ver1.06　標準価格 ¥4,800

発売中
SX-WINDOW用Photo-CDビュアー
SX-PhotoGallery　基本セット ¥15,000

お求めはお近くのパソコンショップ、または弊社通販部(TEL:0286-22-9811)へお申し込みください。
通販ご希望の方は、ソフト代金＋送料¥1,000に消費税を加え、ご住所・お名前・電話番号・商品名を明記した紙を同封の上、現金封筒でお申し込みください。

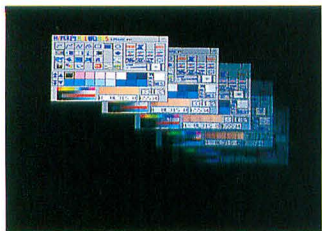
低金利クレジット　通信販売送料 全国一律¥1,000 長期クレジット可能　※表示価格に消費税は含まれておりません

株式会社 計測技研　マイコンショップ BASIC HOUSE 本社／ショールーム／通販部
〒321 栃木県宇都宮市竹林町503-1
TEL 0286-22-9811　FAX 0286-25-3970

インダストリアルマジック

ハイパーピクセルワークス

クイックレスポンスウィンドウシステム

X68000特有のスプライトを利用した高速なウィンドウシステムです。ウィンドウを移動しても書き換えすることなく瞬時に移動することができます。マウスは右ボタンに機能を持たせることによって快適な操作性を実現しています。右クリックによるスポイト機能はもちろん、アイテムの機能を教えてくれるクイックヘルプ機能を装備しています。ウィンドウには、機能をつかいやすく分類してコンパクトにまとめ、描画領域を最大限利用できるように設計してあります。また、リアルタイムバッファ管理などの新技術により2MByteのメモリーでもUNDO機能と２画面データバッファの利用ができます。（2MBytesでは、一部でアンドゥに制限が生じます。ただし、コピーバッファは2MBytesでもフルサイズまでサポートされます。）

アンチエリアスプリミティブレベル対応

プリミティブレベルでアンチエリアスに対応しています。ドットをオーバーサンプリングすることでジャギーのないシャープなエッジのラインを描画できます。15bit colorのもつ表現力を最大限引き出せます。

透明度機能

ほぼすべての機能で透明度を指定できます。ガラスなど向こう側が透けるような表現は、透明度機能を使うことによって簡単に実現できます。

多彩な合成機能

色をつかって描画するかわりに、裏画面を利用できます。多くの機能で裏画面のデータをそのまま利用できますので非常に簡単に、しかも多彩な表現が可能な合成ができます。グラデーションによる合成、透明度を利用した半透明合成などCGならではの表現が可能です。

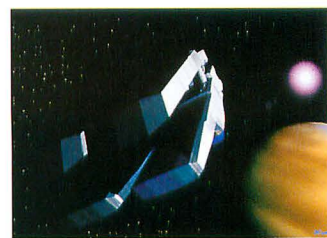
豊富なエフェクト機能

波状変形、モーションブラーなど様々なエフェクトを利用できます。XOR,AND,ORなどの論理演算もサポートし、利用範囲が拡大します。また、まったく新しいアルゴリズムによる細線化処理は、美しく完璧な細線をつくりだします。

マルチマッピング

自在な形状へのマッピングができます。水泡のような不定形への写り込み表現も簡単に実現できます。

エクステンション機能

ハイパーピクセルワークスの最大の特徴が、このエクステンション機能です。個別のプログラムによる機能の拡張ではなく、ハイパーピクセルワークス本体を中核とした一種のウィンドウシステムです。ハイパーピクセルワークスのサポートする機能をすべて継承し、さらに無限の拡張性を提供することができます。周辺機器は、エクステンションによってサポートされます。基本セットでは、グラデーションサークル、逆光などが利用できます。また、マルチフォントエクステンションなど各種ツールが、今後続々と提供されます。

光を自在に操る光学機能

光を表現するための機能として、星、放射光などの機能があります。星をつかうとキラキラと輝く感じを表現できます。放射光をつかうとスポットライトなどの表現を簡単につくりだせます。

４色まで設定できるグラデーション機能

多色グラデーション機能は、最高４色まで設定できます。グラデーションは、内部24bitで描画するので非常に美しい仕上がりを期待できます。

グラフィックツールの未来を予感させるランダムオブジェクト

ランダムオブジェクトをつかうと、グラフィックの自動生成が可能になります。たとえば、木に１枚１枚、葉を描いていくと膨大な作業が必要になります。ランダムオブジェクトを使うと、葉のちらばり具合などを指定すれば自動的に葉を描くことができます。

NSカルコンプ社ドローイングパッドネイティブモード対応

ドローイングパッドをはじめとするCalcompAFT対応タブレット全機種に対応。筆圧、傾きなどを検出できるCalcompAFTにネイティブ対応。

主な機能

［プリミティブ］フリーライン、補間フリーライン、ライン、ベジェカーブ、ボックス、ボックスフィル、サークル、サークルフィル、水平ライン、垂直ライン、正方形、正円、グラデーション、1bitグラデーション、４色グラデーション、ペイント、ポリゴン、アンチエリアスポリゴン、自然色スプレー、1bitスプレー、フラクタルスプレー、プレーンスプレー、スマッジスプレー、ランダムオブジェクト、星、放射光、ROMフォント、全画面ルーペ、［２画面合成］合成ペン、マッピング、マルチマッピング、合成グラデーション、合成1bitグラデーション、合成スプレー、合成ポリゴン、合成ペイント、［マスキング］マスクペン、マスクリムーバーペン、マスクセーブ、マスクロード、［編集］消しゴムペン、UNDO、ペンエディター、拡大縮小、アンチエリアス縮小、自由変形、複写、ダブルカーソルコピー、回転、上下左右反転、タイル生成、色変換８モード、［エフェクト］波状変形、フレア、モーションブラー、エッジ強調、微分（レリーフ）、ぼかし、拡散誤差、強調、AND、XOR、OR、エッジ抽出、1bit変換、プレーン変換、モノクロ、セピア、細線化、［機能］ファイラー、アニメーション、エクステンション、ピクセル1:1view、２画面統合スクロール、［環境設定］ウィンドウオープン設定、データディレクトリ、マウス移動量、フォントパス、RGB-HLS切り替え、タブレット筆圧調整

基本セットがサポートする周辺機器：EPSON GT6000/GT6500、OMRON HS7R、NSカルコンプCalcompAFT対応タブレット全機種

好評発売中

TAKERU価格 ¥19,800（税込） ■X68000,030シリーズ/要2Mバイト ■企画・制作/ハイパー

ハイパーアクション／プロレスカードゲーム

レッスルエンジェルス２

トップイベンター

火花散る四角いリング!! 闘う天使達よ頂点を目指せ!!

★新人レスラーを育ててトップイベンターを目指せ！

マイティ祐希子のIWWFヘビー初戴冠より５年、日本女子マット界の勢力地図は一新された。そんな中、入門したての２人の新人、武藤めぐみと結城千種は明日のチャンピオンを目指して日夜腕をみがいていた。そして、いつしかこの２人もレスラーとしての選択を迫られていく……。

★白熱のVSモード！燃える対戦バトルでファイヤーしろ！

システムはおなじみのカードバトル！技が流れるマルチスポットモーションも健在だ！シングル・タッグ・アドベンチャーの各モードの他、ついにVSモードの登場で対戦プレイが可能に！もちろん恐怖の水着はぎデスマッチもパワーアップ！

TAKERU価格 ¥4,900（税込）

■X68000,030シリーズ/要2Mバイト ■企画・制作/グレイト

B-Field!

魔法文化の世界、ビーフィールド。ある日、王妃が国王を石に変え、国宝の「幻帝の鏡」を持ち去った。賞金目当てにディルズと弟子のレイディは探索の旅に出た。

- ●貧乏学校の師弟コンビが繰り広げるコミカル４択クイズバトルゲーム！
- ●ゲーム・アニメ・漫画、科学、歴史など500問！
- ●通常ストーリーモードの他に２人対戦モード、クイズのみを楽しむチャレンジモードと３種類のモードを用意。
- ●美しいビジュアルシーンには２画面CG収録。

TAKERU価格 ¥3,900（税込） **好評発売中**

■ X68000,030シリーズ ■制作/studioインフェルノ

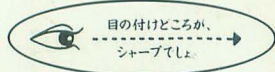
パソコンソフト自販機 TAKERU

ブラザー工業株式会社
〒467 名古屋市瑞穂区苗代町2番1号

TAKERU事務局 (052)824-2493

東京営業所(03)5203−7133
大阪営業所(06)258-3024

通信販売 1994年4月1日より、送料/手数料が有料になります。

ソフト名、機種名、メディアのサイズ、住所、氏名、電話番号を明記の上TAKERU事務局まで現金書留でお申し込みください。送料／手数料は、１回のお申し込み総金額が5,000円以上の方は無料。4,900円までの方は500円をいただきます。4,900円までの方は現金500円をプラスしてお申し込みください。誠に勝手ながら、皆様のご理解とご協力の程、お願い申し上げます。(実施日'94年4月1日より)

ピュア32bitMC68EC030搭載。
クリエイティブパワーが花開くX68030シリーズ。

X68030

本体＋キーボード＋マウス・トラックボール
130㎜FD(5.25型)タイプ
CZ-500C-B(チタンブラック)標準価格398,000円(税別)
HD内蔵 CZ-510C-B(チタンブラック)標準価格488,000円(税別)

X68030 Compact

本体＋キーボード＋マウス
90㎜FD(3.5型)タイプ
CZ-300C-B(チタンブラック)標準価格388,000円(税別)
HD内蔵 CZ-310C-B(チタンブラック)標準価格478,000円(税別)

●写真のカラーディスプレイは別売です。

なか身は、どちらも32ビット。

プロセッサの未来を先取、洗練されたアーキテクチャを誇るMPU MC68000シリーズを搭載。
先駆のクリエイティブ・アビリティで使う人の創造性に応える68ワールドへ、どうぞ。

●消費税及び配送・設置・付帯工事費、使用済み商品の引き取り費等は、標準価格には含まれておりません。

●お問い合わせは…
シャープ株式会社　コンシューマーセンター西日本相談室〒545大阪市阿倍野区長池町22番22号☎(06)621-1221(大代表)　電子機器事業本部システム機器営業部〒545大阪市阿倍野区長池町22番22号☎(06)621-1221(大代表)

T1002179040602 雑誌 02179-4